

Nonlinear Programming

ISE 5406 Course Notes

Dr. Robert Hildebrand

Grado Department of Industrial & Systems Engineering
Virginia Tech

February 3, 2026

Contents

Preface: How to Use These Notes	xv
I Foundations	1
1 Introduction and Mathematical Formulation	3
1.1 Mathematical Formulation and Modeling	3
1.1.1 Example: A Problem for the Ages	4
1.1.2 Example: Production Planning Problem	5
1.2 The Universe is Nonlinear	5
1.3 Machine Learning as Optimization	6
1.3.1 Empirical Risk Minimization	7
1.3.2 Linear Regression and Least Squares	7
1.3.3 LASSO: Sparse Linear Regression	8
1.3.4 Logistic Regression for Classification	8
1.3.5 Support Vector Machines	8
1.3.6 Deep Learning and Neural Networks	9
1.3.7 Summary: ML Problems as Nonlinear Programs	9
1.4 Types of Optimization Problems	10
1.4.1 Deterministic Optimization	10
1.4.2 Optimization Under Uncertainty	10
1.5 Mathematical Optimization: General Formulation	10
1.5.1 Linear and Integer Linear Programs	11
1.5.2 Nonlinear Programming (NLP)	11
1.6 Example: Facility Location Problem	13
1.6.1 Problem Data (Parameters)	13
1.6.2 Decision Variables	13
1.6.3 Objective Function	13
1.6.4 Constraints	14

1.7	Optimality Conditions	14
1.8	Fundamentals of Unconstrained Optimization	14
1.8.1	Example: Least-Squares Problem	15
1.9	Local and Global Minimizers	16
2	Convex Sets and Linear Algebra Fundamentals	19
2.1	Linear Algebra Review	20
2.1.1	Euclidean Space, Hyperplanes, and Halfspaces	20
2.1.2	Types of Combinations	20
2.1.3	Subspaces and Independence	22
2.2	Convex Sets	23
2.2.1	Examples of Convex Sets	24
2.2.2	Operations Preserving Convexity of Sets	26
2.3	Convex and Concave Functions	27
2.3.1	Examples of Convex and Concave Functions	28
2.3.2	Level Sets of Convex Functions	28
2.4	Convex Hulls	29
2.4.1	Carathéodory's Theorem	30
2.5	Introduction to Convex Analysis	30
2.5.1	Extreme Points	30
2.5.2	Rays	31
2.5.3	Recession Directions	31
3	Real Analysis Fundamentals	33
3.1	Closed, Open, and Compact Sets	33
3.2	Operations Preserving Closedness and Openness	37
3.3	Infimum and Supremum	38
3.4	Existence of Minimum and Maximum	39
3.4.1	Bolzano–Weierstrass Theorem	39
3.4.2	Weierstrass' Theorem	40
3.5	Separating Hyperplanes	41
3.5.1	Closest-Point Theorem	41
3.5.2	Separating Hyperplane Theorem	43
3.6	Supporting Hyperplanes	44
3.6.1	Supporting Hyperplane Theorem	45
<i>— End of Verified Material —</i>		46

II Convex Analysis	47
4 Convex Functions and Subgradients	49
4.1 Convex and Concave Functions	50
4.1.1 Strictly Convex and Strictly Concave Functions	51
4.1.2 Examples of Convex and Concave Functions	51
4.2 Operations Preserving Convexity	52
4.3 Continuity of Convex Functions	54
4.4 Characterizations of Differentiable Convex Functions	55
4.4.1 Calculus Background	55
4.4.2 First-Order Condition for Convexity	61
4.4.3 Second-Order Condition for Convexity	62
4.4.4 Running Example: Three Proofs of Convexity	63
4.5 Epigraph and Hypograph	65
4.5.1 Epigraph as the Link Between Convex Sets and Functions	66
4.6 Subgradients	67
4.6.1 Existence of Subgradients	68
4.7 Convex Optimization	69
4.7.1 Optimality Conditions	70
4.7.2 Applications of Optimality Conditions	71
4.7.3 Existence of Optimal Solutions	73
4.7.4 Ordinary Least Squares Regression	73
5 Subgradients and Optimality Conditions	75
5.1 Existence of Subgradients of Convex Functions	75
5.2 Convex Optimization Problems	76
5.3 Necessary and Sufficient Conditions for a Global Minimum .	77
5.4 Applying the Necessary and Sufficient Optimality Conditions	78
5.5 Existence of Optimal Solutions	79
5.6 Ordinary Least Squares Regression	80
5.7 Differentiable Convex Functions	80
5.7.1 The Gradient of a Multivariate Function	80
5.7.2 Characterization of Differentiable Convex Functions .	81
5.7.3 Directional Derivatives of Convex Functions	81
5.7.4 Taylor Series Approximations	82
5.8 Twice Differentiable Convex Functions	82
5.8.1 Twice Differentiable Functions	82
5.8.2 Positive Semidefinite Matrices	83

5.8.3	Characterization of Twice Differentiable Convex Functions	85
5.8.4	Examples of Convex and Concave Functions	86
6	Differentiable Convex Functions	87
6.1	The Gradient of a Multivariate Function	87
6.2	Characterization of Differentiable Convex Functions	88
6.3	Directional Derivatives of Convex Functions	89
6.4	Taylor Series Approximations	90
6.4.1	Univariate Taylor Series	90
6.4.2	Multivariate Taylor Series	90
6.5	Twice Differentiable Functions	91
6.6	Positive Semidefinite Matrices	92
6.6.1	Checking Positive Semidefiniteness	93
6.7	Characterization of Twice Differentiable Convex Functions	94
6.8	Examples of Convex and Concave Functions	95
6.8.1	Examples of Convex Functions	95
6.8.2	Examples of Concave Functions	95
6.9	Recap: Minimizing a Convex Function	95
6.9.1	Necessary and Sufficient Conditions for a Global Minimum	96
6.9.2	Examples	96
7	Generalizations of Convexity	99
7.1	Convex Maximization	99
7.1.1	Necessary Conditions for Local Maxima	100
7.1.2	Extreme Point Solutions	100
7.2	Quasiconvex and Quasiconcave Functions	101
7.2.1	Lower Level Sets of Quasiconvex Functions	102
7.2.2	Quasiconvex Maximization Has an Extreme Point Solution	103
7.3	Strictly Quasiconvex Functions	104
7.4	Strongly Quasiconvex Functions	105
7.5	Strongly Convex Functions	107
7.6	Other Generalizations of Convexity	108
7.6.1	Pseudoconvex Functions	108
7.6.2	Log-Convex Functions	109
7.6.3	Other Generalizations	109
7.7	Summary of Function Classes	110

III Unconstrained Optimization	113
8 Unconstrained Optimization: Conditions and Algorithms 115	
8.1 First-Order Necessary Conditions	115
8.1.1 Descent Directions	116
8.1.2 The First-Order Necessary Condition	117
8.2 Second-Order Necessary and Sufficient Conditions	118
8.2.1 Second-Order Necessary Conditions	118
8.2.2 Second-Order Sufficient Conditions	119
8.2.3 Optimality for Pseudoconvex Functions	119
8.3 Overview of Algorithms for Unconstrained Optimization	121
8.3.1 Two Main Algorithmic Frameworks	121
8.3.2 The Line Search Approach	121
8.4 Line Search Methods	122
8.4.1 Line Search for Strictly Quasiconvex Objectives	122
8.4.2 Line Search Using Derivatives	127
8.4.3 Comparison of Line Search Methods	128
8.5 Summary	131
9 The Steepest Descent Method 133	
9.1 Introduction to the Steepest Descent Method	134
9.2 Exact Line Search	135
9.2.1 Algorithm Description	136
9.2.2 Convergence Analysis	137
9.3 Application to Convex Quadratic Problems	139
9.3.1 Closed-Form Step Size	139
9.3.2 Global Convergence Analysis	140
9.4 Convergence Rate Analysis: A Modern Perspective	143
9.4.1 Key Assumptions: Smoothness and Strong Convexity	143
9.4.2 Convergence Rates for Gradient Descent	145
9.4.3 Summary of Convergence Rates	149
9.5 Inexact Line Search	150
9.5.1 The Sufficient Decrease Condition (Armijo's Rule) .	150
9.5.2 The Wolfe Conditions	151
9.5.3 Backtracking Line Search	153
9.5.4 Numerical Examples	154
9.6 Summary	154

10 Newton's Method and Quasi-Newton Methods	157
10.1 Newton's Method: Motivation and Derivation	157
10.1.1 Derivation of the Newton Direction	158
10.2 Application to Convex Quadratic Programs	158
10.3 Numerical Example	160
10.4 Convergence Properties	161
10.4.1 Potential Issues with Convergence	162
10.4.2 Local Quadratic Convergence	162
10.5 Advantages and Disadvantages	164
10.5.1 Main Advantage	164
10.5.2 Main Disadvantages and Solutions	164
10.6 Guaranteeing Descent with Line Search	164
10.6.1 The Newton Direction as a Descent Direction	165
10.6.2 Newton's Method with Line Search	165
10.7 Levenberg-Marquardt Modification	166
10.7.1 The Modified Hessian	166
10.7.2 The Levenberg-Marquardt Update	166
10.7.3 Interpolation Between Methods	167
10.8 Quasi-Newton Methods	168
10.8.1 Motivation	168
10.8.2 The Secant Condition	169
10.8.3 Popular Quasi-Newton Update Formulae	169
10.8.4 Computational Comparison: Newton vs. BFGS	171
10.9 Summary	171
11 Conjugate Direction and Conjugate Gradient Methods	173
11.1 Conjugate Directions	173
11.1.1 Definition and Basic Properties	174
11.2 Conjugate Direction Method for Convex Quadratic Programs	176
11.2.1 Problem Setup	176
11.2.2 Algorithm Description	176
11.2.3 Derivation of the Step Length	176
11.3 Properties of the Conjugate Direction Method	177
11.3.1 Orthogonality of Gradients to Previous Directions	177
11.3.2 Invariance of Step Lengths	178
11.3.3 Expanding Subspace Minimization	179
11.4 Convergence of the Conjugate Direction Method	180
11.5 Numerical Example	181
11.6 The Conjugate Gradient Method	181
11.6.1 Algorithm Description	182

11.6.2	Derivation of the Direction Update	182
11.7	Nonlinear Conjugate Gradient Methods	183
11.7.1	Motivation	184
11.7.2	Practical Modifications	184
11.7.3	Nonlinear CG Algorithm	185
12	Modern First-Order Methods	187
12.1	Stochastic Gradient Descent	187
12.1.1	Motivation: Finite-Sum Problems	188
12.1.2	The SGD Algorithm	188
12.1.3	Step Size Selection	189
12.1.4	Convergence Analysis	190
12.1.5	Mini-Batch SGD	190
12.2	Momentum Methods	191
12.2.1	Polyak's Heavy Ball Method	191
12.2.2	Nesterov's Accelerated Gradient	191
12.2.3	Convergence Rates for Accelerated Methods	192
12.2.4	Momentum in Deep Learning	193
12.3	Proximal Gradient Methods	193
12.3.1	Composite Optimization Problems	194
12.3.2	The Proximal Operator	194
12.3.3	The Proximal Gradient Algorithm	196
12.3.4	Convergence of Proximal Gradient	196
12.3.5	Accelerated Proximal Gradient (FISTA)	198
12.4	Summary and Connections	198
IV	Constrained Optimization	201
13	Equality-Constrained Optimization	203
13.1	Problem Setup and Regularity Assumptions	203
13.1.1	Problem Formulation	204
13.1.2	The Jacobian Matrix	204
13.1.3	Regular Points	204
13.2	First-Order Necessary Conditions: The Lagrange Multiplier Theorem	205
13.2.1	The Lagrangian Function	206
13.3	Interpretations of the Lagrange Multiplier Theorem	206
13.3.1	Gradient in Constraint Gradient Space	208
13.3.2	Orthogonality to Feasible Variations	208

13.3.3 Stationary Points of the Lagrangian	208
13.3.4 Using the First-Order Conditions	210
13.4 Numerical Examples	210
13.5 Failure of Regularity	212
13.6 Convex Programs with Equality Constraints	213
13.6.1 Problem Formulation	213
13.7 Specialization to Convex Quadratic Programs	214
13.7.1 Problem Formulation	214
13.7.2 Solving via the Lagrange Conditions	214
13.8 Second-Order Optimality Conditions	215
13.8.1 The Subspace of First-Order Feasible Variations	215
13.8.2 The Hessian of the Lagrangian	216
13.8.3 Second-Order Necessary Conditions	216
13.8.4 Second-Order Sufficient Conditions	217
13.9 Numerical Example of Second-Order Conditions	217
13.10 Summary	219
14 Algorithms for Constrained Optimization	221
14.1 Overview of Optimization Models and Software	222
14.1.1 Problem Classes	222
14.1.2 Modeling Languages and Solvers	223
14.2 Algorithms for Constrained Optimization: Overview	223
14.2.1 Taxonomy of Algorithms	224
14.3 The Projected Gradient Method	224
14.3.1 Motivation and Basic Framework	224
14.4 Computing Projections	225
14.4.1 Projection for Box Constraints	225
14.4.2 Projected Gradient Method for Box-Constrained Problems	226
14.4.3 Projection for Linear Equality Constraints	227
14.4.4 Projected Gradient Method for Equality-Constrained Problems	228
14.5 Penalty Methods	229
14.5.1 Motivation	229
14.5.2 Quadratic Penalty Method	230
14.5.3 Nonsmooth (Exact) Penalty Method	230
14.5.4 Penalty Methods for General Constraints	231
14.6 Augmented Lagrangian Methods	231
14.6.1 The Augmented Lagrangian Function	231
14.6.2 The Augmented Lagrangian Algorithm	232

14.7 Active Set Methods	233
14.7.1 Active Constraints and the Active Set	233
14.7.2 Key Idea	233
14.7.3 Sequential Linear Programming (SLP)	234
14.7.4 Sequential Quadratic Programming (SQP)	234
14.8 Interior-Point Methods	235
14.8.1 The Barrier Approach	235
14.8.2 Interpretation	235
14.8.3 Convergence	236
14.9 Summary and Further Reading	236
15 KKT Conditions for Inequality-Constrained Problems	239
15.1 Geometric Necessary Conditions for Constrained Problems	239
15.1.1 Cones of Feasible and Improving Directions	240
15.1.2 Geometric Necessary Condition	241
15.2 Problems with Inequality Constraints	241
15.2.1 Necessary Condition for Inequality-Constrained Problems	244
15.2.2 Numerical Example	244
15.2.3 Issues with the Geometric Necessary Condition	245
15.3 Fritz John Necessary Optimality Conditions	245
15.3.1 Terminology for Fritz John Conditions	247
15.3.2 Numerical Examples for Fritz John Conditions	248
15.4 Potential Issues with the Fritz John Conditions	251
15.4.1 Trivial Satisfaction of FJ Conditions	252
15.4.2 FJ Points Can Be Nonoptimal for Linear Programs	252
15.5 Karush-Kuhn-Tucker (KKT) Necessary Conditions	252
15.5.1 Statement of the KKT Conditions	253
15.5.2 Terminology for KKT Conditions	254
15.5.3 Numerical Examples for KKT Conditions	254
15.6 KKT Conditions for Linear Programs	255
15.7 Geometric Interpretation of the KKT Conditions	256
15.7.1 Insights into the KKT Conditions	257
15.8 Constraint Qualifications	258
15.8.1 Alternative Constraint Qualifications	258
15.9 KKT Conditions for Convex Optimization	259
15.9.1 KKT Conditions Are Not Always Necessary for Convex Problems	259
15.9.2 KKT Conditions Are Sufficient for Convex Problems	261
15.10 Summary	262

16 KKT Conditions: Mixed Constraints and Summary	265
16.1 Review: KKT Necessary Conditions for Inequality Constraints	265
16.2 Problems with Both Inequality and Equality Constraints	266
16.2.1 Key Sets for Mixed Constraints	266
16.3 Geometric Necessary Optimality Condition	267
16.4 KKT Necessary Conditions for Mixed Constraints	269
16.4.1 Understanding the KKT Conditions for Mixed Constraints	269
16.5 Numerical Examples	270
16.6 KKT Sufficient Conditions for Convex Problems	274
16.7 KKT Second-Order Conditions	274
16.7.1 The Restricted Lagrangian and Critical Cone	275
16.7.2 Second-Order Necessary Conditions	276
16.7.3 Second-Order Sufficient Conditions	276
16.8 Examples: Applying Second-Order Conditions	277
16.9 Course Overview and Synthesis	280
16.9.1 Types of Optima	280
16.9.2 Linear Algebra Foundations	280
16.9.3 Real Analysis Foundations	280
16.9.4 Convexity Theory	280
16.9.5 Generalizations of Convexity	281
16.9.6 Convex Analysis	281
16.9.7 Properties of Convex Optimization Problems	281
16.9.8 Optimality Conditions	281
16.9.9 Algorithms for Unconstrained Optimization	282
16.9.10 Algorithms for Constrained Optimization	282
Introduction to Proof Writing	283
.1 Why Learn Proofs?	283
.2 What is a Proof?	284
.3 Common Proof Techniques	284
.3.1 Direct Proof	284
.3.2 Proof by Contraposition	284
.3.3 Proof by Contradiction	285
.3.4 Counterexample	285
.4 Other Proof Techniques	286
.5 Quantifiers	286
.5.1 Using Quantifiers in Proofs	287
.6 Tips for Writing Clear Proofs	287
.7 In-Class Exercise	288

Introduction to Proofs in Real Analysis	289
.8 Classifying Sets: Open, Closed, and Compact	289
.9 The Half-Open Box	290
.10 The Unit Ball and Unit Sphere	291
.11 The Unit Box	292
.12 A Lower-Dimensional Set	292
.13 Useful Theorems for Proving Openness and Closedness	292
Proving or Disproving Convexity of Sets and Functions	295
.14 Proving Convexity of Sets	295
.15 Disproving Convexity of Sets	297
.16 Proving Convexity of Functions	298
.17 Disproving Convexity of Functions	299
.18 Operations Preserving Convexity	300

Preface: How to Use These Notes

Welcome to ISE 5406: Nonlinear Programming! These course notes have been designed to serve as a study companion throughout the semester, summarizing the key material covered in lectures. Here is some guidance on how to get the most out of this document.

Course Learning Objectives

Having successfully completed this course, you will be able to:

1. **Derive analytical techniques** for characterizing structural properties and Fritz John and KKT optimality conditions for nonlinear programming problems, including unconstrained optimization, constrained optimization, and convex optimization problems.
2. **Evaluate the operation and performance** of different numerical algorithms such as line search, steepest descent, Newton's method, conjugate directions, projection gradient, and affine scaling methods for solving nonlinear optimization problems.
3. **Apply knowledge** pertaining to applications of nonlinear programming (theory and algorithms) for solving problems arising in applications such as machine learning and portfolio management.

Structure of the Notes

These notes are organized into four parts that build upon each other:

Part I: Foundations covers the essential mathematical background you will need, including convex sets, linear algebra fundamentals, and real

analysis concepts. If you find yourself struggling with later material, returning to these chapters often helps.

Part II: Convex Analysis develops the theory of convex functions, subgradients, and optimality conditions for convex optimization problems. This material forms the theoretical core of the course.

Part III: Unconstrained Optimization presents algorithms for solving optimization problems without constraints, including steepest descent, Newton's method, and conjugate gradient methods.

Part IV: Constrained Optimization extends the theory to problems with equality and inequality constraints, culminating in the celebrated Karush-Kuhn-Tucker (KKT) conditions.

Important: These Notes Are Not a Substitute for the Textbooks

These notes are intended to *accompany* your study of the course material, not to replace the reference textbooks. While this document covers the core lecture content, **you are expected to do reading beyond this material**. There are several important reasons to consult the textbooks directly:

- The textbooks contain additional results, alternative proofs, and deeper discussions that may not appear here.
- Working through different presentations of the same material reinforces your understanding.
- The reference texts for this course—particularly Bazaraa et al. and Nocedal & Wright—are **standard references in the field of optimization**. Familiarity with these books will serve you well in future coursework, research, and professional work.
- The textbooks contain many more examples and exercises than can be covered in class.

Each chapter begins with a **Recommended Reading** box that lists the corresponding sections from the course textbooks. You are *strongly encouraged* to read both these notes and the textbook sections for a complete understanding.

A Note on the Development of These Notes

Important Disclaimer

This document was largely developed using AI tools, based on prior lecture slides and course material. While the content has been reviewed and appears to be largely correct, **please exercise caution** in relying solely on these notes:

- Question any assumptions or proofs that seem unclear or incomplete.
- Cross-reference important results with the textbooks.
- If you find errors or have questions, please let the professor know.

That said, much of this material is “standard” in the optimization literature, so the presentation should be reliable. If there is further content covered in class that is not documented here and that you would like expanded upon, please let the professor know and these notes can be updated.

Copyright Notice — Do Not Distribute

This document is for ISE 5406 course use only and should not be shared beyond this class.

This material is largely based on the course textbooks and includes graphics and tables borrowed directly from:

- Bazaraa, Sherali, and Shetty, *Nonlinear Programming: Theory and Algorithms*
- Nocedal and Wright, *Numerical Optimization*
- Boyd and Vandenberghe, *Convex Optimization*

Due to copyright restrictions, this document may not be redistributed, posted online, or shared with individuals outside of this course. Please respect the intellectual property of the original authors.

How to Read Each Chapter

Throughout the notes, you will encounter several types of highlighted environments:

- **Definitions** (pink boxes) introduce new terminology and concepts.
- **Theorems and Lemmas** (maroon/orange boxes) state the main results.
- **Examples** (green boxes) illustrate concepts with concrete problems.
- **Remarks** (gray boxes) provide additional insights, warnings, or connections to other topics.

Study Strategies

1. **Before class:** Skim the relevant chapter to familiarize yourself with the main concepts and notation.
2. **After class:** Read the chapter carefully, working through all examples with pencil and paper. Mathematics is not a spectator sport—you learn by doing!
3. **Consult the textbooks:** After reading these notes, read the corresponding sections in the reference texts. Pay attention to alternative explanations, additional examples, and exercises.
4. **Pay attention to proofs:** Understanding *why* a result is true is just as important as knowing *what* the result says. The proof techniques you learn here will help you solve new problems.
5. **Work the exercises:** The homework problems assigned from the textbook are essential for mastering the material. These notes provide the theory; the exercises develop your problem-solving skills.
6. **Form study groups:** Explaining concepts to others is one of the best ways to solidify your own understanding.

Reference Textbooks

Lectures are primarily based on the first two reference texts listed below. All are standard references in the field.

1. **Bazaraa, M.S., Sherali, H.D., and Shetty, C.M.** (2006). *Nonlinear Programming: Theory and Algorithms*, 3rd Edition. John Wiley & Sons.
Our primary textbook, with comprehensive coverage of theory.
E-book available at: <https://onlinelibrary-wiley-com.ezproxy.lib.vt.edu/doi/book/10.1002/0471787779>
2. **Nocedal, J. and Wright, S.J.** (2006). *Numerical Optimization*, 2nd Edition. Springer.
Excellent for algorithmic aspects and numerical methods.
E-book available at: <https://link.springer.com/book/10.1007/978-0-387-40065-5>
3. **Boyd, S.P. and Vandenberghe, L.** (2004). *Convex Optimization*. Cambridge University Press.
A modern treatment of convex optimization.
Freely available at: <https://web.stanford.edu/~boyd/cvxbook/>
4. **Bertsekas, D.P.** (2016). *Nonlinear Programming*, 3rd Edition. Athena Scientific.
An alternative perspective with excellent treatment of duality.
Supplementary material at: <http://www.athenasc.com/nonlinbook.html>

Supplementary Texts for Modern Methods

The following texts provide coverage of stochastic gradient descent, momentum methods, proximal methods, and other modern first-order methods essential for machine learning applications:

1. **Wright, S.J. and Recht, B.** (2022). *Optimization for Data Analysis*. Cambridge University Press.
Accessible treatment of optimization from a machine learning perspective. Excellent for stochastic gradient descent, momentum methods, and proximal methods.
2. **Bubeck, S.** (2015). *Convex Optimization: Algorithms and Complexity*. Foundations and Trends in Machine Learning, Vol. 8, No. 3-4, pp. 231–357.
Rigorous treatment of convergence rate analysis and complexity theory. Recommended for theoretical depth.
Freely available at: <https://arxiv.org/abs/1405.4980>

A Note on Mathematical Maturity

This course assumes familiarity with linear algebra, multivariable calculus, and basic real analysis. It is *your responsibility* to address any gaps in prerequisite knowledge. If you need to refresh these prerequisites, Appendix A of Boyd and Vandenberghe provides an excellent review.

Remember: optimization is a beautiful subject that combines elegant theory with powerful practical applications. These notes aim to convey both aspects. Enjoy the journey!

*Dr. Robert Hildebrand
Blacksburg, Virginia*

Part I

Foundations

Chapter 1

Introduction and Mathematical Formulation

This chapter introduces the fundamental concepts of mathematical optimization, providing the foundation for the study of nonlinear programming. We begin by discussing the art of mathematical modeling and the importance of balancing model complexity with tractability. We then present the general mathematical formulation of optimization problems and survey the various types of optimization problems encountered in practice, including linear, integer, and nonlinear programming. The chapter concludes with a discussion of local versus global minimizers, a distinction that is central to the theory and algorithms of nonlinear optimization.

Recommended Reading

- Chapter 1 and Section 2.1 of Nocedal and Wright (2006)
- Chapter 1 of Bazaraa, Sherali, and Shetty (2006)
- Chapter 1 of Boyd and Vandenberghe
- **Supplementary:** Chapter 1 of Wright and Recht (2022) — excellent overview of machine learning applications

1.1 Mathematical Formulation and Modeling

Optimization is the maximization or minimization of an **objective** subject to **constraints** on its **variables**.

- **Objective:** e.g., profit, time, cost, etc.
- **Variables (or unknowns):** e.g., production/inventory level, etc.
- **Constraints:** non-negativity, bounds, physical laws, etc.

When constructing a mathematical model, one must balance simplicity and accuracy:

1. A simplistic model yields no insight into the true problem.
2. A complex model may be difficult to solve.

Good modeling is half the solution! (see Section 1.3 of Bazaraa et al.)

For further guidance on mathematical modeling, see Fourer, Gay, and Kernighan (2003), *AMPL: A Modeling Language for Mathematical Programming*, available at <https://vanderbei.princeton.edu/307/textbook/AMPLbook.pdf>.

1.1.1 Example: A Problem for the Ages

Example 1.1. Find the ages of Justin and Brad, given that the sum of their ages equals 76 years and that Brad will be twice as old as Justin in 7 years' time.

Solution: Let x and y denote the ages of Justin and Brad, respectively. Then

$$x + y = 76, \tag{1.1}$$

$$2(x + 7) = y + 7. \tag{1.2}$$

Upon solving these equations, we get $x = 23$ and $y = 53$.

Geometric Interpretation: The solution corresponds to the intersection of two lines in the (x, y) -plane.

1.1.2 Example: Production Planning Problem

Example 1.2 (Production Planning). In a manufacturing firm, given the demand for two days along with per-unit production and inventory costs, decide the production schedule for both days.

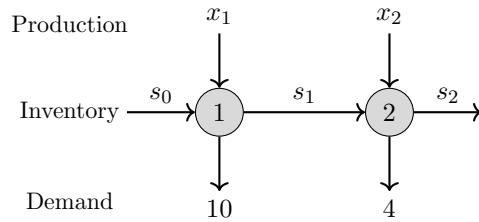


Figure 1.1: Production planning network over two days. Nodes represent days, with production x_i flowing in from above and demand flowing out below. Inventory s_i flows horizontally between days.

Decision Variables:

- x_1 : Production on Day 1
- x_2 : Production on Day 2
- s_0 : Initial Inventory
- s_1 : Inventory at the end of Day 1
- s_2 : Inventory at the end of Day 2

Data:

- Per-unit production cost on Day 1 and Day 2: \$10 and \$16, respectively.
- Per-unit inventory cost: \$5.
- Demand on Day 1: 10 units; Demand on Day 2: 4 units.

1.2 The Universe is Nonlinear

Nonlinear optimization problems arise naturally in many real-world applications, including:

6CHAPTER 1. INTRODUCTION AND MATHEMATICAL FORMULATION

- Electric power grid optimization
- Biological systems and metabolic networks (e.g., *E. coli* modeling)
- Rocket trajectory optimization
- Chemical plant design and operations
- Neural network training

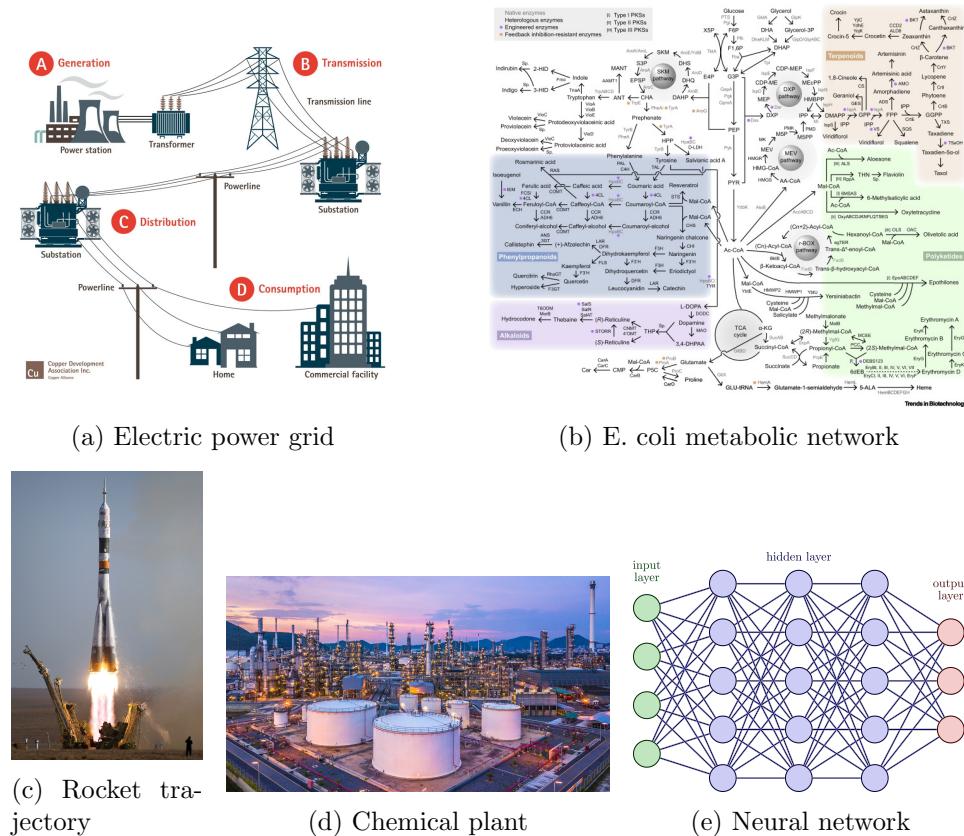


Figure 1.2: Examples of nonlinear optimization applications across various domains.

1.3 Machine Learning as Optimization

Machine learning has emerged as one of the most important application areas for nonlinear optimization. Nearly all machine learning algorithms can be

framed as optimization problems: given training data, find parameters that minimize a **loss function** measuring the discrepancy between predictions and observations. This section introduces the key optimization formulations underlying modern machine learning.

Recommended Reading

Supplementary Reading: For an in-depth treatment of these applications, see Chapter 1 of Wright and Recht (2022), *Optimization for Data Analysis*.

1.3.1 Empirical Risk Minimization

The fundamental paradigm of supervised learning is **empirical risk minimization**. Given training data $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ where $\mathbf{a}_i \in \mathbb{R}^n$ are feature vectors and b_i are labels (or responses), we seek a parameter vector $\mathbf{x} \in \mathbb{R}^n$ that minimizes the average loss:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^m \ell(\mathbf{a}_i^\top \mathbf{x}, b_i),$$

where $\ell(\cdot, \cdot)$ is a loss function measuring prediction error. Different choices of ℓ lead to different machine learning methods.

1.3.2 Linear Regression and Least Squares

For **linear regression** with continuous responses $b_i \in \mathbb{R}$, we use the squared loss:

$$\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2.$$

This gives the classical **least squares** problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2m} \sum_{i=1}^m (\mathbf{a}_i^\top \mathbf{x} - b_i)^2 = \frac{1}{2m} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2,$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ has rows \mathbf{a}_i^\top and $\mathbf{b} = (b_1, \dots, b_m)^\top$. This is a smooth, convex, unconstrained optimization problem with a closed-form solution when \mathbf{A} has full column rank: $\mathbf{x}^* = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$.

1.3.3 LASSO: Sparse Linear Regression

In high-dimensional settings where the number of features n may exceed the number of samples m , we often seek **sparse** solutions. The **LASSO** (Least Absolute Shrinkage and Selection Operator) adds an ℓ_1 penalty to encourage sparsity:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2m} \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_1,$$

where $\lambda > 0$ is a regularization parameter and $\|\mathbf{x}\|_1 = \sum_{j=1}^n |x_j|$. The ℓ_1 penalty induces sparsity by driving many coefficients exactly to zero.

Remark 1.1. The LASSO objective is convex but **nonsmooth** due to the ℓ_1 term. Specialized algorithms such as **proximal gradient methods** (Chapter 12) are designed for such problems.

1.3.4 Logistic Regression for Classification

For binary classification where $b_i \in \{-1, +1\}$, we model the probability of class membership using the logistic (sigmoid) function. The **logistic regression** problem minimizes the cross-entropy loss:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^m \log \left(1 + e^{-b_i \mathbf{a}_i^\top \mathbf{x}} \right).$$

This is a smooth, convex, unconstrained problem. Unlike least squares, it has no closed-form solution and must be solved iteratively using methods such as gradient descent or Newton's method (Chapters 9–10).

1.3.5 Support Vector Machines

The **Support Vector Machine** (SVM) finds a maximum-margin separating hyperplane between classes. The soft-margin SVM formulation is:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^m \max\{0, 1 - b_i \mathbf{a}_i^\top \mathbf{x}\} + \frac{\lambda}{2} \|\mathbf{x}\|_2^2.$$

The first term is the **hinge loss**, which penalizes misclassified points and points within the margin. The second term is a regularizer that controls the complexity of the classifier.

Remark 1.2. The hinge loss $\max\{0, 1 - z\}$ is convex but nonsmooth. Like LASSO, SVMs require techniques for nonsmooth optimization, which we will study later in the course.

1.3.6 Deep Learning and Neural Networks

Deep learning extends these ideas by composing multiple layers of nonlinear transformations. A neural network with L layers computes:

$$\phi(\mathbf{a}; \theta) = \sigma_L(W_L \sigma_{L-1}(W_{L-1} \cdots \sigma_1(W_1 \mathbf{a}) \cdots)),$$

where $\theta = (W_1, \dots, W_L)$ are the weight matrices and σ_ℓ are nonlinear activation functions (e.g., ReLU: $\sigma(z) = \max\{0, z\}$).

Training a neural network means solving:

$$\min_{\theta} \frac{1}{m} \sum_{i=1}^m \ell(\phi(\mathbf{a}_i; \theta), b_i).$$

This is typically a **high-dimensional, nonconvex** optimization problem with millions or billions of parameters. Despite the nonconvexity, **stochastic gradient descent** (Chapter 12) has proven remarkably effective at finding good solutions.

Remark 1.3. The gradient of the loss with respect to all parameters is computed efficiently via **backpropagation**, which is simply the chain rule applied systematically through the network layers.

1.3.7 Summary: ML Problems as Nonlinear Programs

Problem	Convex?	Smooth?	Chapter
Least Squares	Yes	Yes	9–10
LASSO	Yes	No	12
Logistic Regression	Yes	Yes	9–10
SVM (hinge loss)	Yes	No	12
Neural Networks	No	Yes	9, 12

The theory and algorithms developed in this course—gradient descent, Newton’s method, optimality conditions, stochastic methods—form the computational foundation for all of modern machine learning.

1.4 Types of Optimization Problems

There is no *universal* optimization algorithm. There are numerous algorithms, each of which is tailored to a particular type of optimization problem.

1.4.1 Deterministic Optimization

- **Linear Programming** [ISE 5405]
- **Nonlinear Programming** [ISE 5406] — the focus of this course
- **Integer Programming** [ISE 6414]
- **Conic Programming** [ISE 6514]
- **Mixed-Integer Nonlinear Programming** [ISE 6984]

1.4.2 Optimization Under Uncertainty

- **Stochastic and Robust Optimization** [ISE 6454]

1.5 Mathematical Optimization: General Formulation

Definition 1.1 (Optimization Problem). Optimization is the maximization or minimization of a function subject to constraints on its variables.

Mathematically, an optimization problem can be written as:

$$\begin{aligned}
 & \text{Minimize } f(\mathbf{x}) && \text{(objective function)} \\
 & \text{subject to } g_i(\mathbf{x}) \leq 0, \quad i \in \mathcal{I}, && \text{(inequality constraints)} \\
 & \quad h_i(\mathbf{x}) = 0, \quad i \in \mathcal{E}, && \text{(equality constraints)} \\
 & \quad \mathbf{x} \in X \subset \mathbb{R}^n,
 \end{aligned}$$

where f , g_i , and h_i are scalar-valued functions of variables \mathbf{x} , and \mathcal{I} , \mathcal{E} are finite sets of indices.

Depending on the structure of (f, g, h, X) , the problem may be classified as a linear program (LP), mixed-integer linear program, convex program, nonlinear program (NLP), mixed-integer nonlinear program (MINLP), stochastic program, bilevel program, etc.

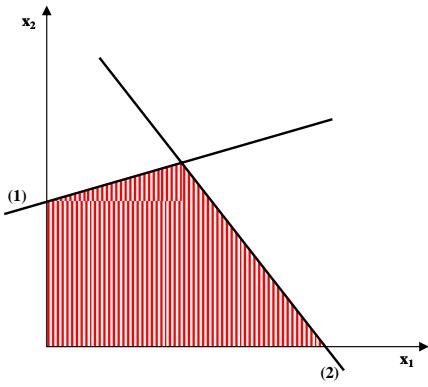
1.5.1 Linear and Integer Linear Programs

Linear Program (LP):

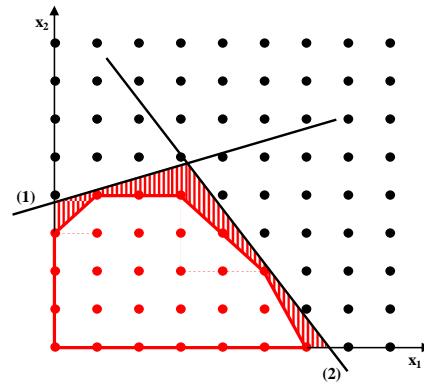
$$\max\{\mathbf{c}^\top \mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \in \mathbb{R}^n\}$$

Integer Linear Program (ILP):

$$\max\{\mathbf{c}^\top \mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \text{ integral}\}$$



(a) LP solution space (continuous)



(b) IP solution space (integer points)

Figure 1.3: Comparison of feasible regions for linear and integer programs.

For integer programs, the **Cutting Plane Algorithm** is a fundamental technique. **Cutting planes** are linear inequalities that cut off part of the relaxed LP feasible region without eliminating any integer-feasible solutions. The strongest cuts are called **facets**.

1.5.2 Nonlinear Programming (NLP)

Example 1.3 (Constrained NLP).

$$\begin{aligned} \min \quad & (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{s.t.} \quad & x_1^2 - x_2 \leq 0, \\ & x_1 + x_2 \leq 2. \end{aligned}$$

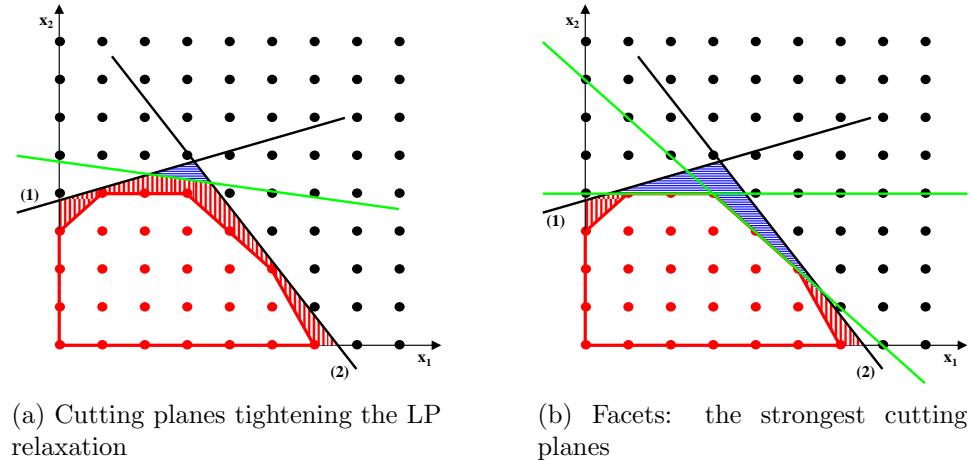


Figure 1.4: The cutting plane approach for integer programming.

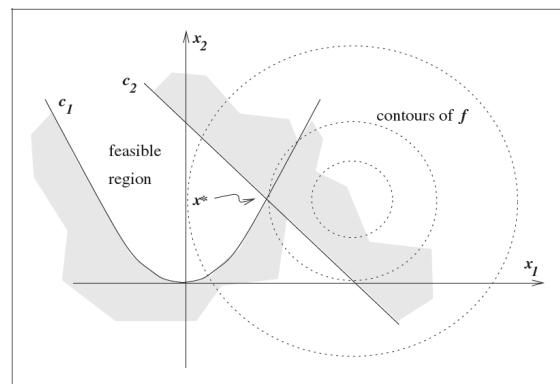


Figure 1.1 Geometrical representation of an optimization problem.

Figure 1.5: Feasible region and contours for the constrained NLP example. The shaded region represents the feasible set defined by the constraints $x_1^2 - x_2 \leq 0$ and $x_1 + x_2 \leq 2$.

Example 1.4 (Unconstrained NLP).

$$\begin{aligned} \min \quad & x_1 + x_1 x_2 + x_2^2 \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{R}^2. \end{aligned}$$

1.6 Example: Facility Location Problem

Suppose that there are n markets with known demands and locations. These demands are to be met from m warehouses of known capacities. The problem is to determine the locations of the warehouses so that the total distance weighted by the shipment from the warehouses to the markets is minimized.

1.6.1 Problem Data (Parameters)

1. c_i : capacity of warehouse i for $i = 1, \dots, m$
2. (a_j, b_j) : known location of market j for $j = 1, \dots, n$
3. r_j : known demand at market j for $j = 1, \dots, n$

1.6.2 Decision Variables

1. (x_i, y_i) : unknown location of warehouse i for $i = 1, \dots, m$
2. w_{ij} and d_{ij} : units shipped and distance, respectively, from warehouse i to market area j for $i = 1, \dots, m; j = 1, \dots, n$

1.6.3 Objective Function

Minimize the total weighted distance:

$$\sum_{i=1}^m \sum_{j=1}^n w_{ij} d_{ij}$$

The distance d_{ij} can be measured using various metrics:

- **Rectilinear Distance:** $d_{ij} = |x_i - a_j| + |y_i - b_j|$
- **Euclidean Distance:** $d_{ij} = \sqrt{(x_i - a_j)^2 + (y_i - b_j)^2}$
- **ℓ_p Norm Metrics:** $d_{ij} = (|x_i - a_j|^p + |y_i - b_j|^p)^{1/p}$

1.6.4 Constraints

1. Capacity constraints:

$$\sum_{j=1}^n w_{ij} \leq c_i, \quad \text{for } i = 1, \dots, m$$

2. Demand constraints:

$$\sum_{i=1}^m w_{ij} = r_j, \quad \text{for } j = 1, \dots, n$$

3. Non-negativity constraints: $w_{ij} \geq 0$ for all i, j .

Remark 1.4. If the locations of the warehouses are fixed, then the d_{ij} values are known constants and this problem reduces to a linear program known as the **transportation problem**.

See Chapter 1 of Bazaraa et al. for more examples of NLP models.

1.7 Optimality Conditions

After an optimization algorithm has been applied to a model, we must be able to recognize whether it has succeeded in finding an optimal solution.

In many cases, there are elegant mathematical expressions, known as **optimality conditions**, for checking that the returned solution is indeed an optimal solution of the problem.

If these optimality conditions are not satisfied, they may provide useful information on how the current estimate of the solution can be improved.

Optimality conditions are closely related to **sensitivity analysis**, which studies the sensitivity of the solution to changes in model parameters.

1.8 Fundamentals of Unconstrained Optimization

Consider the unconstrained optimization problem:

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{Minimize}} \quad f(\mathbf{x}), \quad \text{where } f : \mathbb{R}^n \rightarrow \mathbb{R} \text{ is smooth.}$$

Usually, we lack a global perspective on the function f .

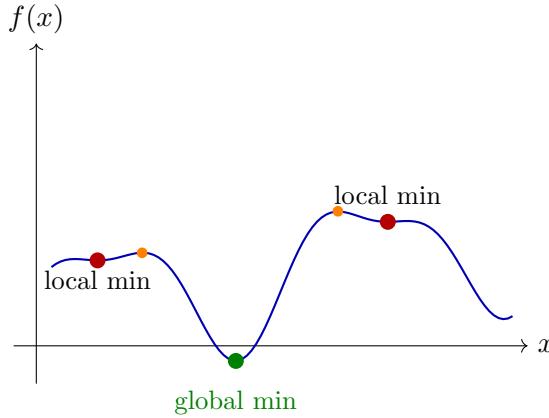


Figure 1.6: A nonconvex function with multiple local minima. Without global knowledge of the function, optimization algorithms may converge to different local minimizers depending on the starting point. The global minimum (green) is the best among all local minima (red).

All we know are the values of f and perhaps some of its derivatives at a set of points $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots$

We wish to design robust algorithms that solve the problem without using too much computer time or storage. Often, the information about f or its derivatives is not cheap to obtain, so we prefer algorithms that do not call for this information unnecessarily.

1.8.1 Example: Least-Squares Problem

Given the measurements y_1, y_2, \dots, y_m of a signal taken at times t_1, t_2, \dots, t_m , find a curve that fits this data.

Suppose we model the relationship between y and t as:

$$\phi(t; \mathbf{x}) = x_1 + x_2 e^{-(x_3 - t)^2/x_4} + x_5 \cos(x_6 t),$$

where real numbers x_i , $i = 1, \dots, 6$, are model parameters.

We wish to choose x_1, x_2, \dots, x_6 so that the model values $\phi(t_j; \mathbf{x})$ fit the observed data y_j , $j = 1, \dots, m$, as closely as possible.

Variables: $\mathbf{x} = (x_1, \dots, x_6) \in \mathbb{R}^6$

Objective function: Minimize the sum of squared residuals, which measure the discrepancy between the model and the observed data:

$$\min f(\mathbf{x}) = r_1^2(\mathbf{x}) + \dots + r_m^2(\mathbf{x}),$$

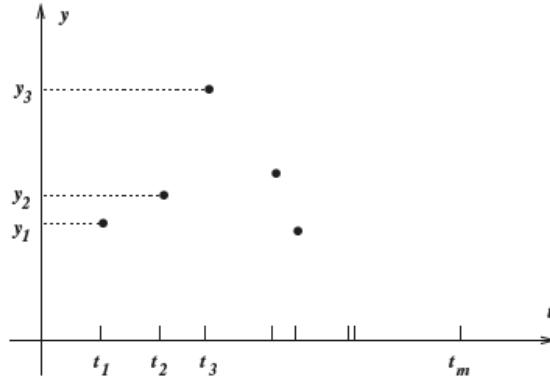


Figure 1.7: Data points and fitted curve for the least-squares problem. The goal is to find model parameters that minimize the sum of squared residuals between the observed data and the model predictions.

where $r_j(\mathbf{x}) = y_j - \phi(t_j, \mathbf{x})$ for $j = 1, \dots, m$.

Chapter 10 of Nocedal and Wright specializes algorithms for unconstrained optimization to least-squares problems.

1.9 Local and Global Minimizers

Definition 1.2 (Global Minimizer). A point \mathbf{x}^* is a **global minimizer** if

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \text{for all } \mathbf{x}.$$

Definition 1.3 (Local Minimizer). A point \mathbf{x}^* is a **local minimizer** if there is some neighborhood \mathcal{N} of \mathbf{x}^* such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{N}.$$

Note that a neighborhood of \mathbf{x}^* is simply an open set that contains \mathbf{x}^* . More precisely, an ϵ -neighborhood of a point $\mathbf{x} \in \mathbb{R}^n$ is the set

$$\mathcal{N}_\epsilon(\mathbf{x}) := \{\mathbf{y} : \|\mathbf{y} - \mathbf{x}\|_2 < \epsilon\}.$$

Definition 1.4 (Strict Local Minimizer). A point \mathbf{x}^* is a **strict** (or **strong**) **local minimizer** if there is some neighborhood \mathcal{N} of \mathbf{x}^* such

that

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

Definition 1.5 (Isolated Local Minimizer). A point \mathbf{x}^* is an **isolated local minimizer** if there is some neighborhood \mathcal{N} of \mathbf{x}^* such that \mathbf{x}^* is the only local minimizer in \mathcal{N} .

Remark 1.5. Strict local minimizers are not always isolated, but all isolated local minimizers are strict. For instance, the point $x^* = 0$ is a strict (but not isolated) local minimizer of the function

$$f(x) = \begin{cases} x^4 \cos(1/x) + 2x^4 & \text{if } x \neq 0, \\ 0 & \text{if } x = 0. \end{cases}$$

Plot this function and see how it looks near $x = 0$! See Figure 3.7 of Bazaraa et al. for other examples.

Reading for the Next Chapter

- Chapter 2 of Bazaraa et al. (2006)
- Sections 2.1 to 2.3 and Sections 3.1 to 3.2 of Boyd and Vandenberghe

Chapter 2

Convex Sets and Linear Algebra Fundamentals

This chapter establishes the foundational concepts of convex sets and the necessary linear algebra background for nonlinear programming. We begin with a review of essential linear algebra definitions including hyperplanes, halfspaces, and various types of combinations (linear, conic, affine, and convex). We then introduce convex sets, examine important examples such as polyhedra, cones, and ellipsoids, and explore operations that preserve convexity. The chapter also provides an introduction to convex and concave functions, convex hulls, Carathéodory's theorem, extreme points, and recession directions. These concepts form the geometric and algebraic foundation upon which the theory of nonlinear optimization is built.

Recommended Reading

- Chapter 2 of Bazaraa, Sherali, and Shetty (2006)
- Parts of Chapters 2 and 3 of Boyd and Vandenberghe

2.1 Linear Algebra Review

2.1.1 Euclidean Space, Hyperplanes, and Halfspaces

Definition 2.1 (Euclidean Space). The **Euclidean space** \mathbb{R}^n is the collection of all vectors of dimension n .

Definition 2.2 (Hyperplane). A **hyperplane** is a set of the form

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{x} = b\},$$

where $\mathbf{a} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ and $b \in \mathbb{R}$.

Remark 2.1. A hyperplane corresponds to a line when $n = 2$ and a plane when $n = 3$. Geometrically, a hyperplane is the set of points \mathbf{x} with constant inner product (b) to a given vector \mathbf{a} . Every hyperplane divides \mathbb{R}^n into two **halfspaces**.

Definition 2.3 (Halfspaces). The two **halfspaces** defined by a hyperplane are

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{x} \leq b\} \quad \text{and} \quad \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{x} \geq b\},$$

where $\mathbf{a} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ and $b \in \mathbb{R}$.

2.1.2 Types of Combinations

Consider vectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ in \mathbb{R}^n .

Definition 2.4 (Linear Combination). A vector $\mathbf{b} \in \mathbb{R}^n$ is said to be a **linear combination** of $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ if

$$\mathbf{b} = \sum_{j=1}^k \lambda_j \mathbf{a}_j,$$

where $\lambda_j \in \mathbb{R}$ for $j = 1, \dots, k$.

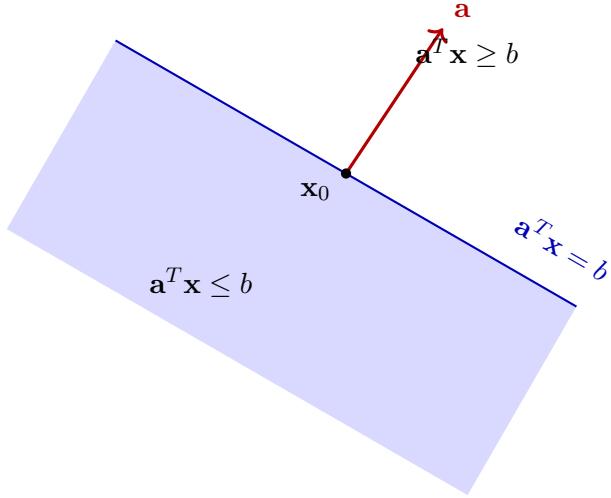


Figure 2.1: Geometric interpretation of halfspaces. The hyperplane $\mathbf{a}^T \mathbf{x} = b$ (blue line) divides \mathbb{R}^n into two halfspaces. The normal vector \mathbf{a} (red arrow) points into the halfspace $\mathbf{a}^T \mathbf{x} \geq b$. The shaded region represents the halfspace $\mathbf{a}^T \mathbf{x} \leq b$.

Definition 2.5 (Conic Combination). A vector $\mathbf{b} \in \mathbb{R}^n$ is said to be a **conic combination** of $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ if

$$\mathbf{b} = \sum_{j=1}^k \lambda_j \mathbf{a}_j,$$

where $\lambda_j \geq 0$ for $j = 1, \dots, k$.

Definition 2.6 (Affine Combination). A vector $\mathbf{b} \in \mathbb{R}^n$ is said to be an **affine combination** of $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ if

$$\mathbf{b} = \sum_{j=1}^k \lambda_j \mathbf{a}_j,$$

where $\lambda_j \in \mathbb{R}$ for $j = 1, \dots, k$, and $\sum_{j=1}^k \lambda_j = 1$.

Definition 2.7 (Convex Combination). A vector $\mathbf{b} \in \mathbb{R}^n$ is said to be a **convex combination** of $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ if

$$\mathbf{b} = \sum_{j=1}^k \lambda_j \mathbf{a}_j,$$

where $\lambda_j \in [0, 1]$ for $j = 1, \dots, k$, and $\sum_{j=1}^k \lambda_j = 1$.

2.1.3 Subspaces and Independence

Definition 2.8 (Linear Subspace). A set $S_L \subseteq \mathbb{R}^n$ is a **linear subspace** if for any $\mathbf{a}_1, \mathbf{a}_2 \in S_L$, every linear combination of \mathbf{a}_1 and \mathbf{a}_2 is also in S_L .

Definition 2.9 (Affine Subspace). A set $S_A \subseteq \mathbb{R}^n$ is an **affine subspace** if for any $\mathbf{a}_1, \mathbf{a}_2 \in S_A$, every affine combination of \mathbf{a}_1 and \mathbf{a}_2 is also in S_A .

Remark 2.2. Every linear subspace is an affine subspace, but the converse is not true. A linear subspace must contain the origin, whereas an affine subspace is a translation of a linear subspace and need not contain the origin.

Definition 2.10 (Linear Independence). Vectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ in \mathbb{R}^n are said to be **linearly independent** if

$$\sum_{j=1}^k \lambda_j \mathbf{a}_j = \mathbf{0} \implies \lambda_j = 0 \text{ for all } j = 1, \dots, k.$$

Definition 2.11 (Affine Independence). Vectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ in \mathbb{R}^n are said to be **affinely independent** if $\mathbf{a}_2 - \mathbf{a}_1, \mathbf{a}_3 - \mathbf{a}_1, \dots, \mathbf{a}_k - \mathbf{a}_1$ are linearly independent.

Equivalently, vectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ are affinely independent if

$$\sum_{j=1}^k \lambda_j = 0 \text{ and } \sum_{j=1}^k \lambda_j \mathbf{a}_j = \mathbf{0} \implies \lambda_j = 0 \text{ for all } j = 1, \dots, k.$$

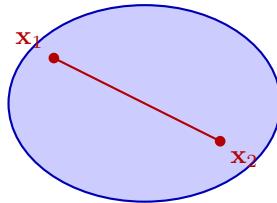
Definition 2.12 (Span). Vectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ in \mathbb{R}^n are said to **span** \mathbb{R}^n if any vector in \mathbb{R}^n can be represented as their linear combination.

2.2 Convex Sets

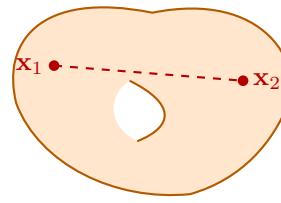
Definition 2.13 (Convex Set). A set $X \subseteq \mathbb{R}^n$ is **convex** if for any two points \mathbf{x}_1 and \mathbf{x}_2 in X and any $\lambda \in [0, 1]$,

$$\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \in X.$$

Remark 2.3 (Geometric Interpretation). For each pair of points \mathbf{x}_1 and \mathbf{x}_2 in X , the line segment joining them must be entirely contained in X .



Convex set



Nonconvex set

Figure 2.2: Illustration of convex and nonconvex sets. The set on the left is convex because every line segment connecting two points in the set remains within the set. The set on the right is nonconvex because there exist points whose connecting line segment passes outside the set (shown as a dashed line).

Example 2.1 (Convexity Examples). Consider which of the following sets are convex:

1. The Euclidean space \mathbb{R}^n and the empty set \emptyset — *both are convex*.
2. $\{\mathbf{x} \in \mathbb{R}^2 : x_1^2 + x_2^2 \leq 1\}$ and $\{\mathbf{x} \in \mathbb{R}^2 : x_1^2 + x_2^2 < 1\}$ — *both are convex* (closed and open disks).
3. $\{\mathbf{x} : \mathbf{Ax} = \mathbf{b}\}$, where \mathbf{A} is an $m \times n$ matrix and $\mathbf{b} \in \mathbb{R}^m$ — *convex* (affine set).
4. $\{\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, where \mathbf{A} is an $m \times n$ matrix and $\mathbf{b} \in \mathbb{R}^m$ — *convex* (polyhedron).
5. Hyperplanes and halfspaces — *convex*.
6. $\{\mathbf{x} \in \mathbb{R}_+^2 : x_1 x_2 = 0\}$ — *not convex* (union of two coordinate axes in the first quadrant).
7. $\{\mathbf{x} \in \mathbb{R}^2 : x_1^2 - x_1 x_2^2 + x_2^4 + 1 \geq 0\}$ — *requires analysis of the constraint*.

2.2.1 Examples of Convex Sets

Polyhedra and Polyhedral Cones

Definition 2.14 (Polyhedral Set / Polyhedron). A **polyhedral set** or **polyhedron** is the intersection of a finite number of halfspaces:

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\},$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$.

Definition 2.15 (Polytope). A **polytope** is a bounded polyhedron.

Remark 2.4. Polyhedral sets are convex. A constraint $\mathbf{a}^T \mathbf{x} \leq \beta$ of a polyhedron is called **geometrically redundant** if removing it does not add any new points to the set.

Definition 2.16 (Polyhedral Cone). A **polyhedral cone** is the intersection of a finite number of halfspaces whose hyperplanes pass through the origin:

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{0}\}.$$

Polyhedral cones are a special case of polyhedral sets.

Definition 2.17 (Convex Cone). A **convex cone** is a convex set X with the additional property that

$$\mathbf{x} \in X \text{ and } \lambda \geq 0 \implies \lambda\mathbf{x} \in X.$$

Polyhedral cones are a special case of convex cones.

Second-Order Cone and Positive Semidefinite Cone

Definition 2.18 (Lorentz or Second-Order Cone). The **Lorentz cone** or **second-order cone** is defined as

$$\mathcal{L}^{n+1} := \{(\mathbf{x}, t) \in \mathbb{R}^n \times \mathbb{R} : \|\mathbf{x}\|_2 \leq t\}.$$

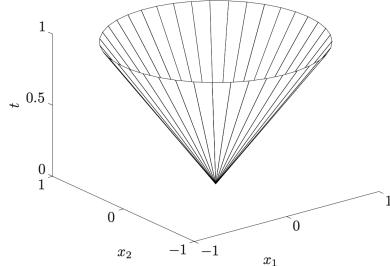


Figure 2.3: The second-order cone (Lorentz cone) in \mathbb{R}^3 . The cone consists of all points (\mathbf{x}, t) where the Euclidean norm of \mathbf{x} is bounded by t , forming an ice-cream cone shape.

Definition 2.19 (Positive Semidefinite Cone). The **positive semidefinite cone** is defined as

$$\mathcal{S}_+^n := \{\mathbf{X} \in \mathcal{S}^n : \mathbf{X} \succeq 0\},$$

where \mathcal{S}^n is the set of symmetric $n \times n$ matrices.

Euclidean Balls and Ellipsoids

Definition 2.20 (Euclidean Ball). The **Euclidean ball** with center \mathbf{x}_c and radius r is defined as

$$\mathcal{B}(\mathbf{x}_c, r) := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{x}_c\|_2 \leq r\}.$$

Definition 2.21 (Ellipsoid). An **ellipsoid** is defined as

$$\mathcal{E} := \{\mathbf{x}_c + \mathbf{A}\mathbf{u} : \|\mathbf{u}\|_2 \leq 1\} \subseteq \mathbb{R}^n,$$

where $\mathbf{A} \in \mathcal{S}_+^n$. If \mathbf{A} is positive semidefinite but not positive definite (i.e., singular), the resulting set is a **degenerate ellipsoid** that lies in a lower-dimensional affine subspace.

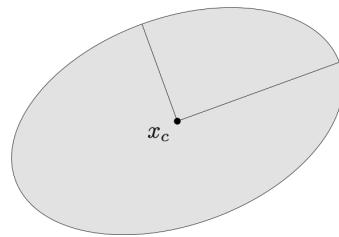


Figure 2.4: An ellipsoid in \mathbb{R}^2 . The ellipsoid is the image of the unit ball under an affine transformation, with center \mathbf{x}_c and shape determined by the matrix \mathbf{A} .

2.2.2 Operations Preserving Convexity of Sets

Lemma 2.1 (Operations Preserving Convexity). *Let S, S_1, S_2 be convex sets in \mathbb{R}^n . Then the following sets are convex:*

1. **Intersection:** $S_1 \cap S_2$.
2. **Minkowski Addition:** $S_1 \oplus S_2 := \{\mathbf{x}_1 + \mathbf{x}_2 : \mathbf{x}_1 \in S_1, \mathbf{x}_2 \in S_2\}$.

3. **Minkowski Difference:** $S_1 \ominus S_2 := \{\mathbf{x}_1 - \mathbf{x}_2 : \mathbf{x}_1 \in S_1, \mathbf{x}_2 \in S_2\}$.
4. **Translation:** $S + a := \{\mathbf{x} + a : \mathbf{x} \in S\}$ for any $a \in \mathbb{R}^n$.
5. **Scaling:** $tS := \{t\mathbf{x} : \mathbf{x} \in S\}$ for any $t \in \mathbb{R}$.
6. **Cartesian Product:** $S_1 \times S_2 := \{(\mathbf{x}_1, \mathbf{x}_2) : \mathbf{x}_1 \in S_1, \mathbf{x}_2 \in S_2\}$.
7. **Coordinate Projection:** $\text{proj}_{\mathbf{x}}(S) := \{\mathbf{x} : (\mathbf{x}, \mathbf{y}) \in S \text{ for some } \mathbf{y}\}$.
8. **Affine Image:** $\{\mathbf{A}\mathbf{x} + \mathbf{b} : \mathbf{x} \in S\}$ for any $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$.
9. **Affine Pre-image:** $\{\mathbf{x} : \mathbf{A}\mathbf{x} + \mathbf{b} \in S\}$ for some $\mathbf{A} \in \mathbb{R}^{n \times m}$, $\mathbf{b} \in \mathbb{R}^n$.

2.3 Convex and Concave Functions

Let $X \subseteq \mathbb{R}^n$ be a convex set.

Definition 2.22 (Convex Function). A function $f : X \rightarrow \mathbb{R}$ is said to be **convex** if for any two points \mathbf{x}_1 and \mathbf{x}_2 in X and any $\lambda \in [0, 1]$,

$$f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda)f(\mathbf{x}_2).$$

Definition 2.23 (Concave Function). A function $f : X \rightarrow \mathbb{R}$ is said to be **concave** if $-f$ is convex. Equivalently, for any two points \mathbf{x}_1 and \mathbf{x}_2 in X and any $\lambda \in [0, 1]$,

$$f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \geq \lambda f(\mathbf{x}_1) + (1 - \lambda)f(\mathbf{x}_2).$$

Remark 2.5. Only affine functions ($\mathbf{a}^T \mathbf{x} + b$) are both convex and concave.

2.3.1 Examples of Convex and Concave Functions

Example 2.2 (Convex Functions). The following are examples of convex functions:

- **Exponential function:** $\exp(x)$ on $X = \mathbb{R}$.
- **Power functions:** x^p on $X = (0, +\infty)$ for $p \geq 1$ or $p \leq 0$.
- **Negative entropy:** $x \ln(x)$ on $X = (0, +\infty)$.
- **Quadratic functions:** $\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$ on $X = \mathbb{R}^n$ for any $\mathbf{A} \in \mathcal{S}_+^n$, $\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$.

Example 2.3 (Concave Functions). The following are examples of concave functions:

- **Power functions:** x^p on $X = (0, +\infty)$ for $p \in [0, 1]$.
- **Geometric mean:** $(\prod_{i=1}^n x_i)^{1/n}$ on $X = \mathbb{R}_{++}^n$.
- **Logarithm:** $\ln(x)$ on $X = (0, +\infty)$.
- **Quadratic functions:** $\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$ on $X = \mathbb{R}^n$ for $-\mathbf{A} \in \mathcal{S}_+^n$, $\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$.

2.3.2 Level Sets of Convex Functions

Definition 2.24 (Level Set). Let S be a convex set and $f : S \rightarrow \mathbb{R}$ be a convex function. For any $\alpha \in \mathbb{R}$, the **α -level set** of f is defined as

$$S_\alpha = \{\mathbf{x} \in S : f(\mathbf{x}) \leq \alpha\}.$$

Lemma 2.2 (Level Sets are Convex). *Let S be a convex set and $f : S \rightarrow \mathbb{R}$ be a convex function. Then for any $\alpha \in \mathbb{R}$, the level set S_α is a convex set.*

Proof. Let $\mathbf{x}_1, \mathbf{x}_2 \in S_\alpha$ and $\lambda \in [0, 1]$. We need to show that $\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \in S_\alpha$.

Since S is convex and $\mathbf{x}_1, \mathbf{x}_2 \in S$, we have $\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \in S$.

Since f is convex:

$$f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2) \leq \lambda\alpha + (1 - \lambda)\alpha = \alpha.$$

Therefore, $\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \in S_\alpha$. \square

Remark 2.6. Upper level sets of concave functions are convex. Specifically, if S is a convex set, $f : S \rightarrow \mathbb{R}$ is a concave function, and $\alpha \in \mathbb{R}$, then $\{\mathbf{x} \in S : f(\mathbf{x}) \geq \alpha\}$ is a convex set.

Remark 2.7. Suppose $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m_I$, are convex and $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, m_E$, are affine. Then

$$\{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0, i = 1, \dots, m_I, h_j(\mathbf{x}) = 0, j = 1, \dots, m_E\}$$

is a convex set. This follows because it is the intersection of level sets of convex functions and affine (hence both convex and concave) functions.

2.4 Convex Hulls

Definition 2.25 (Convex Hull). Let S be an arbitrary set in \mathbb{R}^n . The **convex hull** of S , denoted $\text{conv}(S)$, is the set of all convex combinations of points in S .

In other words, $\mathbf{x} \in \text{conv}(S)$ if and only if it can be represented as

$$\mathbf{x} = \sum_{j=1}^k \lambda_j \mathbf{x}_j, \quad \sum_{j=1}^k \lambda_j = 1, \quad \lambda_j \geq 0, \quad j = 1, \dots, k,$$

for some positive integer k and points $\mathbf{x}_1, \dots, \mathbf{x}_k \in S$.

Definition 2.26 (Polytope). Let $S := \{\mathbf{x}_1, \dots, \mathbf{x}_{k+1}\}$ be a finite set of points in \mathbb{R}^n . A **polytope** is the convex hull of S .

Definition 2.27 (Simplex). If $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{k+1}$ are affinely independent, then $\text{conv}(S)$ is called a **simplex** having vertices $\mathbf{x}_1, \dots, \mathbf{x}_{k+1}$.

Remark 2.8. Any simplex in \mathbb{R}^n can have at most $n+1$ vertices. This is because the maximum number of affinely independent vectors in \mathbb{R}^n is $n+1$ (since the maximum number of linearly independent vectors is n).

Lemma 2.3 (Convex Hull is Smallest Convex Set). *Let S be an arbitrary set in \mathbb{R}^n . Then $\text{conv}(S)$ is the smallest convex set containing S . It is also the intersection of all convex sets containing S .*

2.4.1 Carathéodory's Theorem

Theorem 2.4 (Carathéodory's Theorem). *Let S be an arbitrary set in \mathbb{R}^n . If $\mathbf{x} \in \text{conv}(S)$, then there exist $\mathbf{x}_1, \dots, \mathbf{x}_{n+1} \in S$ such that $\mathbf{x} \in \text{conv}(\mathbf{x}_1, \dots, \mathbf{x}_{n+1})$.*

In other words, there exist $\mathbf{x}_1, \dots, \mathbf{x}_{n+1} \in S$ such that

$$\mathbf{x} = \sum_{j=1}^{n+1} \lambda_j \mathbf{x}_j, \quad \sum_{j=1}^{n+1} \lambda_j = 1, \quad \lambda_j \geq 0, \quad j = 1, \dots, n+1.$$

Remark 2.9. The proof is a reading assignment; see Theorem 2.1.6 of Bazaraa et al. Note that:

- The theorem trivially holds for $\mathbf{x} \in S$.
- The choice of $\mathbf{x}_1, \dots, \mathbf{x}_{n+1}$ can depend on \mathbf{x} .

2.5 Introduction to Convex Analysis

2.5.1 Extreme Points

Definition 2.28 (Extreme Point). Let X be a convex set. A point $\mathbf{x} \in X$ is called an **extreme point** of X if it cannot be represented as a strict convex combination of two distinct points in X .

Mathematically, if $\mathbf{x} = \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2$ for some $\lambda \in (0, 1)$ and $\mathbf{x}_1, \mathbf{x}_2 \in X$, then $\mathbf{x} = \mathbf{x}_1 = \mathbf{x}_2$.

2.5.2 Rays

Definition 2.29 (Ray). A **ray** is a collection of points of the form

$$\{\mathbf{x}_0 + \lambda \mathbf{d} : \lambda \geq 0\},$$

where $\mathbf{d} \neq \mathbf{0}$.

- The point \mathbf{x}_0 is called the **vertex** of the ray.
- The vector \mathbf{d} is called the **direction** of the ray.

Remark 2.10. A ray is a convex set.

2.5.3 Recession Directions

Definition 2.30 (Recession Direction). Let X be a convex set. If for each $\mathbf{x}_0 \in X$, the ray

$$\{\mathbf{x}_0 + \lambda \mathbf{d} : \lambda \geq 0\} \subseteq X,$$

then $\mathbf{d} \neq \mathbf{0}$ is called a **recession direction** of X .

Example 2.4 (Recession Directions). Consider the set

$$X := \{\mathbf{x} \in \mathbb{R}_+^2 : x_1 - x_2 \geq -2, x_1 + x_2 \geq 1\}.$$

This is an unbounded polyhedral set. The recession directions are vectors \mathbf{d} such that starting from any point in X and moving in direction \mathbf{d} , we remain in X . For this set, the recession directions include any non-negative linear combination of vectors pointing along the unbounded edges of the feasible region.

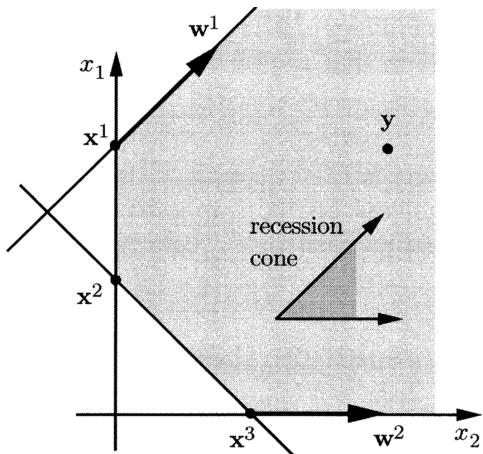


Figure 2.5: Illustration of recession directions for the polyhedral set $X = \{\mathbf{x} \in \mathbb{R}_+^2 : x_1 - x_2 \geq -2, x_1 + x_2 \geq 1\}$. The recession directions form a cone consisting of all directions along which one can move indefinitely while remaining in the set.

Chapter 3

Real Analysis Fundamentals

This chapter establishes the essential concepts from real analysis that form the foundation for optimization theory. We begin with the fundamental notions of closed, open, and compact sets, along with their characterizations through closure, interior, and boundary operations. We then examine how set operations preserve these properties and introduce the concepts of infimum and supremum, which are crucial for analyzing optimization problems. The Weierstrass theorem provides sufficient conditions for the existence of optimal solutions. Finally, we develop the theory of separating and supporting hyperplanes, including the closest-point theorem and Farkas' lemma, which are indispensable tools for characterizing optimality conditions in constrained optimization.

Recommended Reading

- Chapter 2 of Bazaraa, Sherali, and Shetty (2006)
- Chapter 2 and Appendix A of Boyd and Vandenberghe

3.1 Closed, Open, and Compact Sets

Let S be an arbitrary set in \mathbb{R}^n . Most of the following definitions extend naturally beyond \mathbb{R}^n to more general metric spaces.

Definition 3.1 (Closure). A point $\mathbf{x} \in \mathbb{R}^n$ is said to be in the **closure** of S , denoted $\text{cl}(S)$, if $S \cap \mathcal{N}_\epsilon(\mathbf{x}) \neq \emptyset$ for every $\epsilon > 0$, where $\mathcal{N}_\epsilon(\mathbf{x})$ denotes the ϵ -neighborhood of \mathbf{x} .

Definition 3.2 (Interior). A point $\mathbf{x} \in \mathbb{R}^n$ is said to be in the **interior** of S , denoted $\text{int}(S)$, if $\mathcal{N}_\epsilon(\mathbf{x}) \subset S$ for *some* $\epsilon > 0$. A set S is called a **solid set** if it has $\text{int}(S) \neq \emptyset$.

Definition 3.3 (Closed Set). If $S = \text{cl}(S)$, then S is called **closed**.

Definition 3.4 (Open Set). If $S = \text{int}(S)$, then S is called **open**.

Definition 3.5 (Bounded Set). A set S is said to be **bounded** if it can be contained in a ball of sufficiently large radius centered at the origin.

Definition 3.6 (Compact Set). A set S is **compact** if it is *both* closed and bounded.

Definition 3.7 (Boundary). A point $\mathbf{x} \in \mathbb{R}^n$ is said to be on the **boundary** of S , denoted ∂S , if for *every* $\epsilon > 0$, $\mathcal{N}_\epsilon(\mathbf{x})$ contains *at least* one point in S and *at least* one point *not* in S .

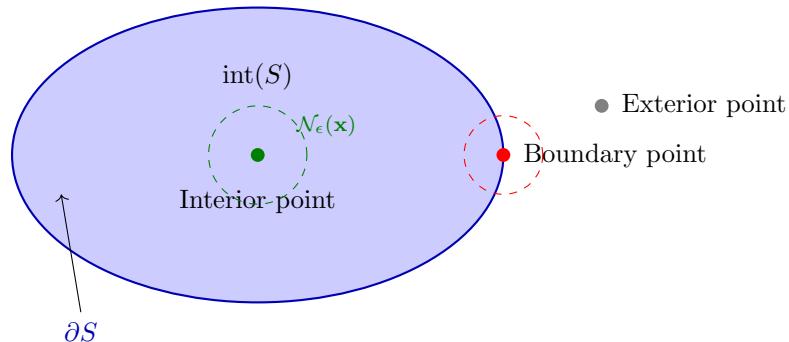


Figure 3.1: Illustration of interior, boundary, and exterior points. An interior point (green) has a neighborhood entirely contained in S . A boundary point (red) has every neighborhood containing points both inside and outside S . An exterior point (gray) lies outside the closure of S .

The following lemmas provide useful characterizations of closed and open

sets.

Lemma 3.1 (Closed Sets). *Suppose $S \subseteq \mathbb{R}^n$ is a closed set. Then:*

1. *S contains all its boundary points, i.e., $\partial S \subseteq S$.*
2. *Its complement $\mathbb{R}^n \setminus S$ is open.*
3. *For any convergent sequence $\{\mathbf{x}_k\}$ in S with $\lim_{k \rightarrow \infty} \mathbf{x}_k = \bar{\mathbf{x}}$, we have $\bar{\mathbf{x}} \in S$.*

Proof. **Part 1:** Let $\mathbf{x} \in \partial S$. By definition of boundary, for every $\epsilon > 0$, the neighborhood $\mathcal{N}_\epsilon(\mathbf{x})$ contains at least one point in S . Thus $S \cap \mathcal{N}_\epsilon(\mathbf{x}) \neq \emptyset$ for all $\epsilon > 0$. By definition of closure, $\mathbf{x} \in \text{cl}(S)$. Since S is closed, $S = \text{cl}(S)$, so $\mathbf{x} \in S$.

Part 2: Let $\mathbf{x} \in \mathbb{R}^n \setminus S$ (i.e., $\mathbf{x} \notin S$). We want to show there exists $\epsilon > 0$ such that $\mathcal{N}_\epsilon(\mathbf{x}) \subseteq \mathbb{R}^n \setminus S$. Suppose not: for all $\epsilon > 0$, $\mathcal{N}_\epsilon(\mathbf{x}) \cap S \neq \emptyset$. By definition of closure, $\mathbf{x} \in \text{cl}(S)$. Since S is closed, $\text{cl}(S) = S$, so $\mathbf{x} \in S$, which is a contradiction. Therefore such $\epsilon > 0$ exists, and $\mathbb{R}^n \setminus S$ is open.

Part 3: Let $\{\mathbf{x}_k\}$ be a sequence in S with $\mathbf{x}_k \rightarrow \bar{\mathbf{x}}$. By definition of convergence, for any $\epsilon > 0$, there exists K such that $\|\mathbf{x}_k - \bar{\mathbf{x}}\| < \epsilon$ for all $k \geq K$. Thus $\mathbf{x}_k \in \mathcal{N}_\epsilon(\bar{\mathbf{x}})$ for $k \geq K$. Since $\mathbf{x}_k \in S$, we have $S \cap \mathcal{N}_\epsilon(\bar{\mathbf{x}}) \neq \emptyset$ for all $\epsilon > 0$. By definition of closure, $\bar{\mathbf{x}} \in \text{cl}(S)$. Since S is closed, $\bar{\mathbf{x}} \in S$. \square

Lemma 3.2 (Open Sets). *A set $S \subseteq \mathbb{R}^n$ is open if and only if it does not contain any of its boundary points, i.e., $\partial S \cap S = \emptyset$.*

Proof. (\Rightarrow) Assume S is open. Let $\mathbf{x} \in S$. Since S is open, $S = \text{int}(S)$, so $\mathbf{x} \in \text{int}(S)$. By definition of interior, there exists $\epsilon > 0$ with $\mathcal{N}_\epsilon(\mathbf{x}) \subseteq S$. Thus $\mathcal{N}_\epsilon(\mathbf{x})$ contains no points outside S . By definition of boundary, $\mathbf{x} \notin \partial S$. Since this holds for all $\mathbf{x} \in S$, we have $\partial S \cap S = \emptyset$.

(\Leftarrow) Assume $\partial S \cap S = \emptyset$. Let $\mathbf{x} \in S$. Since $\mathbf{x} \notin \partial S$ and $\mathbf{x} \in S \subseteq \text{cl}(S)$, and using the key fact that $\text{cl}(S) = \text{int}(S) \cup \partial S$, we must have $\mathbf{x} \in \text{int}(S)$. This holds for all $\mathbf{x} \in S$, so $S \subseteq \text{int}(S)$. Since always $\text{int}(S) \subseteq S$, we have $S = \text{int}(S)$, so S is open. \square

Lemma 3.3 (Boundary). *The boundary of a set $S \subseteq \mathbb{R}^n$ is given by $\partial S = \text{cl}(S) \setminus \text{int}(S)$.*

Proof. We prove both inclusions.

(\subseteq) Let $\mathbf{x} \in \partial S$. First, we show $\mathbf{x} \in \text{cl}(S)$: By definition of boundary, for every $\epsilon > 0$, the neighborhood $\mathcal{N}_\epsilon(\mathbf{x})$ contains at least one point in S , so $S \cap \mathcal{N}_\epsilon(\mathbf{x}) \neq \emptyset$. By definition of closure, $\mathbf{x} \in \text{cl}(S)$.

Next, we show $\mathbf{x} \notin \text{int}(S)$: By definition of boundary, for every $\epsilon > 0$, the neighborhood $\mathcal{N}_\epsilon(\mathbf{x})$ contains at least one point not in S . Thus there is no $\epsilon > 0$ with $\mathcal{N}_\epsilon(\mathbf{x}) \subseteq S$. By definition of interior, $\mathbf{x} \notin \text{int}(S)$.

Therefore $\mathbf{x} \in \text{cl}(S) \setminus \text{int}(S)$.

(\supseteq) Let $\mathbf{x} \in \text{cl}(S) \setminus \text{int}(S)$. From $\mathbf{x} \in \text{cl}(S)$: for every $\epsilon > 0$, $S \cap \mathcal{N}_\epsilon(\mathbf{x}) \neq \emptyset$, so every neighborhood contains a point in S . From $\mathbf{x} \notin \text{int}(S)$: for every $\epsilon > 0$, $\mathcal{N}_\epsilon(\mathbf{x}) \not\subseteq S$, so every neighborhood contains a point not in S . By definition of boundary, $\mathbf{x} \in \partial S$. \square

Example 3.1 (Classifying Sets). Consider the following sets and their properties:

1. The entire space \mathbb{R}^n and the empty set \emptyset are both open and closed (they are the only sets with this property). Neither is compact since \mathbb{R}^n is unbounded.
2. The set $[\mathbf{0}, \mathbf{1}] \subset \mathbb{R}^n$ (including $\mathbf{0}$ but not $\mathbf{1}$) is neither open nor closed. Its interior is $(\mathbf{0}, \mathbf{1})$.
3. The unit ball $S = \{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 \leq 1\}$ is closed and bounded, hence compact. Its interior is the open ball.
4. The unit ball excluding its boundary $S = \{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 < 1\}$ is open but not closed, hence not compact.
5. The unit sphere $S = \{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 = 1\}$ is closed and bounded, hence compact. Its interior is empty.
6. The unit box $S = \left\{ \mathbf{x} \in \mathbb{R}^n : \max_{i \in [n]} |x_i| \leq 1 \right\}$ is closed and bounded, hence compact.
7. The set $S = \{\mathbf{x} \in \mathbb{R}^3 : -1 \leq x_1, x_2 \leq 1, x_3 = 0\}$ is closed and bounded in \mathbb{R}^3 , hence compact. Its interior (in \mathbb{R}^3) is empty.
8. The hyperplane $S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{x} = b\}$ for some $\mathbf{a} \in \mathbb{R}^n$, $b \in \mathbb{R}$, is closed but unbounded (unless $\mathbf{a} = \mathbf{0}$), hence not compact. Its interior is empty.

9. The halfspace $S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{x} \leq b\}$ for some $\mathbf{a} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$, $b \in \mathbb{R}$, is closed but unbounded, hence not compact.
10. The polyhedron $S = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ for some $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, is closed but may or may not be bounded.

3.2 Operations Preserving Closedness and Openness

The following lemma describes how set operations affect the closedness and openness of sets.

Lemma 3.4 (Set Operations and Closedness/Openness).

1. *The intersection of any (even infinite) number of closed sets is closed.*
2. *The intersection of finitely many open sets is open.*
3. *The union of finitely many closed sets is closed.*
4. *The union of any (even infinite) number of open sets is open.*

Remark 3.1. The following assertions are *false* in general:

1. The intersection of infinitely many open sets is open.
2. The union of infinitely many closed sets is closed.

As a counterexample for (1), consider $\bigcap_{n=1}^{\infty} \left(-\frac{1}{n}, \frac{1}{n}\right) = \{0\}$, which is closed but not open. As a counterexample for (2), consider $\bigcup_{n=1}^{\infty} [\frac{1}{n}, 1] = (0, 1]$, which is neither open nor closed.

Theorem 3.5 (Convex Combinations of Closure and Interior Points). *Let S be a convex set in \mathbb{R}^n with a nonempty interior. Let $\mathbf{x}_1 \in \text{cl}(S)$ and $\mathbf{x}_2 \in \text{int}(S)$. Then*

$$\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \in \text{int}(S) \text{ for each } \lambda \in (0, 1).$$

The proof is left as a reading assignment (see Theorem 2.2.2 of Bazaraa et al.).

Corollary 3.6. Let S be a convex set. Then:

1. $\text{int}(S)$ is convex.
2. If $\text{int}(S) \neq \emptyset$, then $\text{cl}(S)$ is convex.
3. If $\text{int}(S) \neq \emptyset$, then $\text{cl}(\text{int}(S)) = \text{cl}(S)$.
4. If $\text{int}(S) \neq \emptyset$, then $\text{int}(\text{cl}(S)) = \text{int}(S)$.

Remark 3.2. We can replace $\text{int}(S)$ with the relative interior of S in the above results.

3.3 Infimum and Supremum

Let S be an arbitrary set in \mathbb{R} .

Definition 3.8 (Infimum). A scalar $l \in \mathbb{R}$ is called the **infimum** or the **greatest lower bound** of S , denoted $\inf(S)$, if:

1. l is a lower bound of S , i.e., $l \leq x$ for each $x \in S$,
2. for all lower bounds b of S , $l \geq b$.

By convention, $\inf(\emptyset) = +\infty$, and $\inf(S) = -\infty$ if S is unbounded from below.

Definition 3.9 (Supremum). A scalar $u \in \mathbb{R}$ is called the **supremum** or the **least upper bound** of S , denoted $\sup(S)$, if:

1. u is an upper bound of S , i.e., $u \geq x$ for each $x \in S$,
2. for all upper bounds b of S , $u \leq b$.

By convention, $\sup(\emptyset) = -\infty$, and $\sup(S) = +\infty$ if S is unbounded from above.

3.4 Existence of Minimum and Maximum

Let $S \subseteq \mathbb{R}^n$ be a nonempty set and $f : S \rightarrow \mathbb{R}$. A natural question in optimization is: When do the problems

$$\inf_{\mathbf{x} \in S} f(\mathbf{x}) / \min_{\mathbf{x} \in S} f(\mathbf{x}) \quad \text{and} \quad \sup_{\mathbf{x} \in S} f(\mathbf{x}) / \max_{\mathbf{x} \in S} f(\mathbf{x})$$

admit (optimal) solutions?

Definition 3.10 (Minimizing and Maximizing Solutions). •

A point $\bar{\mathbf{x}}$ is a **minimizing solution** for the problem $\inf\{f(\mathbf{x}) : \mathbf{x} \in S\}$, provided that $\bar{\mathbf{x}} \in S$ and $f(\bar{\mathbf{x}}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in S$. If a minimizing solution exists, we simply write $\min_{\mathbf{x} \in S} f(\mathbf{x})$.

- A point $\bar{\mathbf{x}}$ is a **maximizing solution** for the problem $\sup\{f(\mathbf{x}) : \mathbf{x} \in S\}$, provided that $\bar{\mathbf{x}} \in S$ and $f(\bar{\mathbf{x}}) \geq f(\mathbf{x})$ for all $\mathbf{x} \in S$. If a maximizing solution exists, we simply write $\max_{\mathbf{x} \in S} f(\mathbf{x})$.

3.4.1 Bolzano–Weierstrass Theorem

The Bolzano–Weierstrass theorem is a fundamental result that underlies the existence of optimal solutions.

Theorem 3.7 (Bolzano–Weierstrass Theorem). *Every bounded sequence in \mathbb{R}^n has a convergent subsequence.*

Proof. We prove this for $n = 1$ using the bisection method. The general case follows by applying this to each coordinate and using a diagonal argument.

Let $\{x_k\}$ be a bounded sequence in \mathbb{R} . Since it is bounded, there exists M such that $|x_k| \leq M$ for all k . Thus all terms lie in $[a_0, b_0] = [-M, M]$.

Construction: Bisect $[a_0, b_0]$ into two halves. At least one half contains infinitely many terms of $\{x_k\}$. Call this half $[a_1, b_1]$ (length = M). Pick $x_{k_1} \in [a_1, b_1]$.

Repeat: Bisect $[a_1, b_1]$, choose the half with infinitely many terms, call it $[a_2, b_2]$ (length = $M/2$). Pick $x_{k_2} \in [a_2, b_2]$ with $k_2 > k_1$.

Continue: Get nested intervals $[a_j, b_j]$ with length $\frac{2M}{2^j} \rightarrow 0$. Pick $x_{k_j} \in [a_j, b_j]$ with $k_1 < k_2 < k_3 < \dots$.

Identifying the limit: By the Nested Interval Theorem, $\bigcap_{j=1}^{\infty} [a_j, b_j] = \{\bar{x}\}$ for some $\bar{x} \in \mathbb{R}$.

Convergence: For any $\epsilon > 0$, choose J large enough that $\frac{2M}{2^J} < \epsilon$. For $j \geq J$: $x_{k_j} \in [a_j, b_j]$ and $\bar{x} \in [a_j, b_j]$, so $|x_{k_j} - \bar{x}| \leq b_j - a_j = \frac{2M}{2^j} < \epsilon$. Therefore $x_{k_j} \rightarrow \bar{x}$.

For \mathbb{R}^n : apply to each coordinate, then use a diagonal argument to extract a common subsequence converging in all coordinates. \square

3.4.2 Weierstrass' Theorem

The following theorem provides sufficient conditions for the existence of optimal solutions.

Theorem 3.8 (Weierstrass' Theorem). *Let S be a nonempty, compact (closed and bounded) set, and let $f : S \rightarrow \mathbb{R}$ be continuous on S . Then the problem $\min\{f(\mathbf{x}) : \mathbf{x} \in S\}$ attains its minimum.*

Proof. Let $\gamma = \inf\{f(\mathbf{x}) : \mathbf{x} \in S\}$. Since S is nonempty, this infimum exists in $[-\infty, \infty)$. By the definition of infimum, for each positive integer k there exists $\mathbf{x}_k \in S$ such that $\gamma \leq f(\mathbf{x}_k) < \gamma + 1/k$, which implies $f(\mathbf{x}_k) \rightarrow \gamma$ as $k \rightarrow \infty$.

Since S is bounded, the sequence $\{\mathbf{x}_k\}$ is bounded. By the Bolzano–Weierstrass theorem (Theorem 3.7), there exists a convergent subsequence $\{\mathbf{x}_{k_j}\}$ with limit $\bar{\mathbf{x}}$. Since S is closed and each $\mathbf{x}_{k_j} \in S$, Lemma 3.1 implies that $\bar{\mathbf{x}} \in S$.

Finally, the continuity of f and the convergence $\mathbf{x}_{k_j} \rightarrow \bar{\mathbf{x}}$ yield $f(\mathbf{x}_{k_j}) \rightarrow f(\bar{\mathbf{x}})$. Since also $f(\mathbf{x}_{k_j}) \rightarrow \gamma$ (as a subsequence of $\{f(\mathbf{x}_k)\}$), uniqueness of limits gives $f(\bar{\mathbf{x}}) = \gamma$. In particular, γ is finite, and $\bar{\mathbf{x}}$ is a minimizer of f over S . \square

Remark 3.3. The conditions in Weierstrass' theorem are sufficient but not necessary. The theorem can fail when:

- S is not closed (e.g., minimizing $f(x) = x$ over $(0, 1)$).
- f is not continuous (e.g., f has a jump discontinuity at the candidate minimum).
- f is unbounded over an unbounded set S (e.g., minimizing $f(x) = x$ over $\{x : x \geq a\}$).

3.5 Separating Hyperplanes

Let S_1 and S_2 be nonempty sets in \mathbb{R}^n .

Definition 3.11 (Types of Separation). Consider a hyperplane $H := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{p}^T \mathbf{x} = \alpha\}$.

- The hyperplane H **separates** S_1 and S_2 if $\mathbf{p}^T \mathbf{x} \geq \alpha$ for each $\mathbf{x} \in S_1$ and $\mathbf{p}^T \mathbf{x} \leq \alpha$ for each $\mathbf{x} \in S_2$.
- If, additionally, $S_1 \cup S_2 \not\subset H$, then H **properly separates** S_1 and S_2 .
- The hyperplane H **strictly separates** S_1 and S_2 if $\mathbf{p}^T \mathbf{x} > \alpha$ for each $\mathbf{x} \in S_1$ and $\mathbf{p}^T \mathbf{x} < \alpha$ for each $\mathbf{x} \in S_2$.
- The hyperplane H **strongly separates** S_1 and S_2 if for some $\epsilon > 0$, $\mathbf{p}^T \mathbf{x} \geq \alpha + \epsilon$ for each $\mathbf{x} \in S_1$ and $\mathbf{p}^T \mathbf{x} \leq \alpha - \epsilon$ for each $\mathbf{x} \in S_2$.

3.5.1 Closest-Point Theorem

Theorem 3.9 (Closest-Point Theorem). *Given a nonempty closed convex set $S \subseteq \mathbb{R}^n$ and a point $\mathbf{y} \in \mathbb{R}^n$ with $\mathbf{y} \notin S$, there exists a unique point $\bar{\mathbf{x}} \in S$ with minimum distance from \mathbf{y} . Moreover, $\bar{\mathbf{x}}$ is the minimizing point if and only if*

$$(\mathbf{y} - \bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) \leq 0, \quad \text{for all } \mathbf{x} \in S.$$

Proof. **Part 1 (Existence):** Define $f(\mathbf{x}) = \|\mathbf{y} - \mathbf{x}\|^2$ (continuous). Since S may be unbounded, we cannot apply Weierstrass directly. Pick any $\mathbf{x}_0 \in S$ and let $r = \|\mathbf{y} - \mathbf{x}_0\|$. Define $S' = S \cap \overline{B}(\mathbf{y}, r)$, the intersection of S with the closed ball of radius r centered at \mathbf{y} .

The set S' is nonempty (contains \mathbf{x}_0), closed (intersection of closed sets), and bounded (subset of $\overline{B}(\mathbf{y}, r)$), hence compact. By Weierstrass' theorem, there exists $\bar{\mathbf{x}} \in S'$ minimizing f over S' .

For any $\mathbf{x} \in S \setminus S'$, we have $\|\mathbf{y} - \mathbf{x}\| > r \geq \|\mathbf{y} - \bar{\mathbf{x}}\|$. Thus $\bar{\mathbf{x}}$ minimizes f over all of S .

Part 2 (Uniqueness): Suppose $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$ are both closest to \mathbf{y} with $d = \|\mathbf{y} - \bar{\mathbf{x}}_1\| = \|\mathbf{y} - \bar{\mathbf{x}}_2\|$. Consider $\hat{\mathbf{x}} = \frac{1}{2}\bar{\mathbf{x}}_1 + \frac{1}{2}\bar{\mathbf{x}}_2$. Since S is convex, $\hat{\mathbf{x}} \in S$.

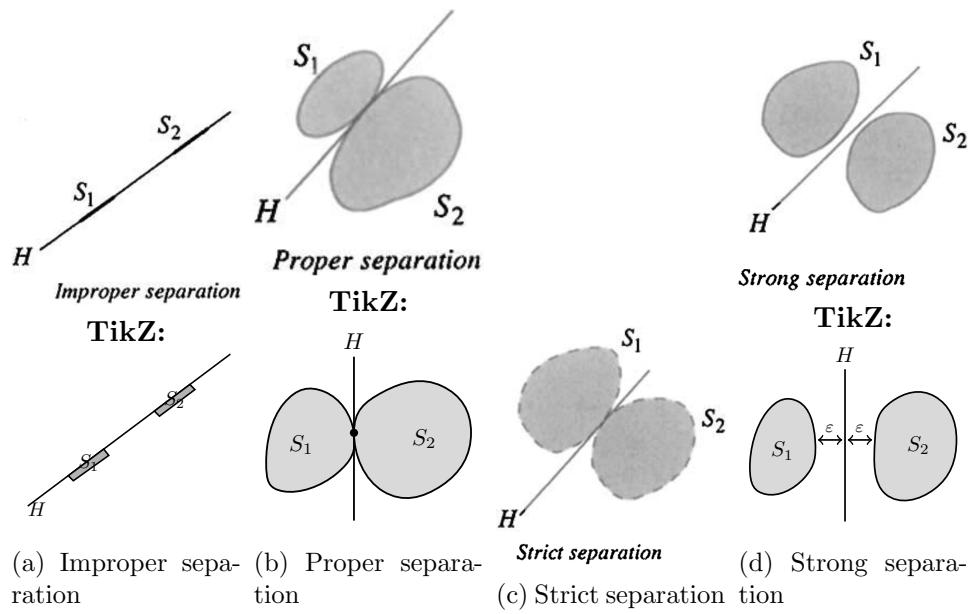


Figure 3.2: Types of hyperplane separation between two sets S_1 and S_2 . In improper separation, both sets may lie entirely within the hyperplane. Proper separation requires at least one set to have points not on the hyperplane. Strict separation requires all points of each set to be strictly on opposite sides. Strong separation guarantees a positive gap between the sets and the hyperplane.

By strict convexity of $\|\cdot\|^2$:

$$\|\mathbf{y} - \hat{\mathbf{x}}\|^2 = \left\| \frac{(\mathbf{y} - \bar{\mathbf{x}}_1) + (\mathbf{y} - \bar{\mathbf{x}}_2)}{2} \right\|^2 < \frac{1}{2}\|\mathbf{y} - \bar{\mathbf{x}}_1\|^2 + \frac{1}{2}\|\mathbf{y} - \bar{\mathbf{x}}_2\|^2 = d^2,$$

unless $\bar{\mathbf{x}}_1 = \bar{\mathbf{x}}_2$. This contradicts the minimality of d , so $\bar{\mathbf{x}}_1 = \bar{\mathbf{x}}_2$.

Part 3 (\Rightarrow): Assume $\bar{\mathbf{x}}$ minimizes distance. Let $\mathbf{x} \in S$ be arbitrary. For $\lambda \in [0, 1]$, define $\mathbf{x}_\lambda = \bar{\mathbf{x}} + \lambda(\mathbf{x} - \bar{\mathbf{x}}) = (1 - \lambda)\bar{\mathbf{x}} + \lambda\mathbf{x}$. Since S is convex, $\mathbf{x}_\lambda \in S$.

Since $\bar{\mathbf{x}}$ minimizes distance: $\|\mathbf{y} - \bar{\mathbf{x}}\|^2 \leq \|\mathbf{y} - \mathbf{x}_\lambda\|^2$ for all $\lambda \in [0, 1]$.

Expanding:

$$\begin{aligned} \|\mathbf{y} - \mathbf{x}_\lambda\|^2 &= \|\mathbf{y} - \bar{\mathbf{x}} - \lambda(\mathbf{x} - \bar{\mathbf{x}})\|^2 \\ &= \|\mathbf{y} - \bar{\mathbf{x}}\|^2 - 2\lambda(\mathbf{y} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) + \lambda^2\|\mathbf{x} - \bar{\mathbf{x}}\|^2. \end{aligned}$$

Thus $0 \leq -2\lambda(\mathbf{y} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) + \lambda^2\|\mathbf{x} - \bar{\mathbf{x}}\|^2$. Dividing by $\lambda > 0$ and letting $\lambda \rightarrow 0^+$: $(\mathbf{y} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \leq 0$.

Part 4 (\Leftarrow): Assume $(\mathbf{y} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \leq 0$ for all $\mathbf{x} \in S$. For any $\mathbf{x} \in S$:

$$\begin{aligned} \|\mathbf{y} - \mathbf{x}\|^2 &= \|(\mathbf{y} - \bar{\mathbf{x}}) - (\mathbf{x} - \bar{\mathbf{x}})\|^2 \\ &= \|\mathbf{y} - \bar{\mathbf{x}}\|^2 - 2(\mathbf{y} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) + \|\mathbf{x} - \bar{\mathbf{x}}\|^2. \end{aligned}$$

Since $(\mathbf{y} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \leq 0$, we have $-2(\mathbf{y} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0$. Thus $\|\mathbf{y} - \mathbf{x}\|^2 \geq \|\mathbf{y} - \bar{\mathbf{x}}\|^2$, so $\bar{\mathbf{x}}$ minimizes distance. \square

The condition $(\mathbf{y} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \leq 0$ has a geometric interpretation: the angle between the vector from $\bar{\mathbf{x}}$ to \mathbf{y} and any vector from $\bar{\mathbf{x}}$ to another point in S must be at least 90 degrees.

3.5.2 Separating Hyperplane Theorem

Theorem 3.10 (Separation of a Convex Set and a Point). *Let S be a nonempty closed convex set in \mathbb{R}^n and $\mathbf{y} \notin S$. Then, there exists a nonzero vector \mathbf{p} and a scalar α such that*

$$\mathbf{p}^T \mathbf{y} > \alpha \quad \text{and} \quad \mathbf{p}^T \mathbf{x} \leq \alpha \text{ for each } \mathbf{x} \in S.$$

Proof. **Step 1:** Since S is nonempty, closed, and convex, and $\mathbf{y} \notin S$, by the Closest-Point Theorem (Theorem 3.9), there exists a unique $\bar{\mathbf{x}} \in S$ closest to \mathbf{y} , and $(\mathbf{y} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \leq 0$ for all $\mathbf{x} \in S$.

Step 2: Define $\mathbf{p} = \mathbf{y} - \bar{\mathbf{x}}$. Note that $\mathbf{p} \neq \mathbf{0}$ since $\mathbf{y} \neq \bar{\mathbf{x}}$ (because $\mathbf{y} \notin S$ but $\bar{\mathbf{x}} \in S$).

Define $\alpha = \mathbf{p}^T \bar{\mathbf{x}} = (\mathbf{y} - \bar{\mathbf{x}})^T \bar{\mathbf{x}}$.

Step 3 (Verify $\mathbf{p}^T \mathbf{x} \leq \alpha$ for all $\mathbf{x} \in S$): From the Closest-Point condition: $(\mathbf{y} - \bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) \leq 0$. Thus $\mathbf{p}^T \mathbf{x} - \mathbf{p}^T \bar{\mathbf{x}} \leq 0$, which gives $\mathbf{p}^T \mathbf{x} \leq \mathbf{p}^T \bar{\mathbf{x}} = \alpha$.

Step 4 (Verify $\mathbf{p}^T \mathbf{y} > \alpha$):

$$\begin{aligned}\mathbf{p}^T \mathbf{y} &= (\mathbf{y} - \bar{\mathbf{x}})^T \mathbf{y} \\ &= (\mathbf{y} - \bar{\mathbf{x}})^T \bar{\mathbf{x}} + (\mathbf{y} - \bar{\mathbf{x}})^T (\mathbf{y} - \bar{\mathbf{x}}) \\ &= \alpha + \|\mathbf{y} - \bar{\mathbf{x}}\|^2.\end{aligned}$$

Since $\mathbf{y} \neq \bar{\mathbf{x}}$, we have $\|\mathbf{y} - \bar{\mathbf{x}}\|^2 > 0$, so $\mathbf{p}^T \mathbf{y} > \alpha$. \square

Corollary 3.11 (Intersection of Halfspaces). *Let S be a closed convex set in \mathbb{R}^n . Then S is the intersection of all halfspaces containing S .*

Corollary 3.12 (Strong Separation). *Let S be a nonempty set, and let $\mathbf{y} \notin \text{cl}(\text{conv}(S))$. Then there exists a strongly separating hyperplane separating S and $\{\mathbf{y}\}$.*

Corollary 3.13 (Farkas' Lemma). *Let $A \in \mathbb{R}^{m \times n}$ and $\mathbf{c} \in \mathbb{R}^n$. Then exactly one of the following two systems has a solution:*

System 1: $A\mathbf{x} \leq \mathbf{0}$ and $\mathbf{c}^T \mathbf{x} > 0$ for some $\mathbf{x} \in \mathbb{R}^n$.

System 2: $A^T \mathbf{y} = \mathbf{c}$ and $\mathbf{y} \geq \mathbf{0}$ for some $\mathbf{y} \in \mathbb{R}^m$.

3.6 Supporting Hyperplanes

Let S be a nonempty set in \mathbb{R}^n and $\bar{\mathbf{x}} \in \partial S$.

Definition 3.12 (Supporting Hyperplane). A hyperplane $H := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{p}^T(\mathbf{x} - \bar{\mathbf{x}}) = 0\}$ is called a **supporting hyperplane** of S at $\bar{\mathbf{x}}$ if S is entirely contained in one of the halfspaces

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{p}^T(\mathbf{x} - \bar{\mathbf{x}}) \leq 0\}, \quad \text{or} \quad \{\mathbf{x} \in \mathbb{R}^n : \mathbf{p}^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0\}.$$

Definition 3.13 (Proper Supporting Hyperplane). If $S \not\subseteq H$, then H is a **proper supporting hyperplane** of S at $\bar{\mathbf{x}}$.

3.6.1 Supporting Hyperplane Theorem

Theorem 3.14 (Supporting Hyperplane Theorem). *Let S be a nonempty convex set in \mathbb{R}^n and $\bar{\mathbf{x}} \in \partial S$. Then there exists a hyperplane that supports S at $\bar{\mathbf{x}}$. That is, there exists $\mathbf{p} \neq \mathbf{0}$ such that $\mathbf{p}^T(\mathbf{x} - \bar{\mathbf{x}}) \leq 0$ for each $\mathbf{x} \in \text{cl}(S)$.*

Proof. Since $\bar{\mathbf{x}} \in \partial S$, every neighborhood of $\bar{\mathbf{x}}$ contains points outside $\text{cl}(S)$. Thus for each positive integer k , there exists $\mathbf{y}_k \notin \text{cl}(S)$ with $\|\mathbf{y}_k - \bar{\mathbf{x}}\| < 1/k$, so $\mathbf{y}_k \rightarrow \bar{\mathbf{x}}$.

Since $\text{cl}(S)$ is a nonempty closed convex set and $\mathbf{y}_k \notin \text{cl}(S)$, the Separating Hyperplane Theorem (Theorem 3.10) guarantees the existence of $\mathbf{p}_k \neq \mathbf{0}$ and α_k such that $\mathbf{p}_k^T \mathbf{y}_k > \alpha_k$ and $\mathbf{p}_k^T \mathbf{x} \leq \alpha_k$ for all $\mathbf{x} \in \text{cl}(S)$. Without loss of generality, normalize so that $\|\mathbf{p}_k\| = 1$.

The sequence $\{\mathbf{p}_k\}$ lies on the unit sphere, which is compact. By the Bolzano–Weierstrass theorem, there exists a convergent subsequence $\mathbf{p}_{k_j} \rightarrow \mathbf{p}$ with $\|\mathbf{p}\| = 1$, so $\mathbf{p} \neq \mathbf{0}$.

For any $\mathbf{x} \in \text{cl}(S)$, the separation condition gives $\mathbf{p}_{k_j}^T \mathbf{x} \leq \alpha_{k_j} < \mathbf{p}_{k_j}^T \mathbf{y}_{k_j}$. Since $\bar{\mathbf{x}} \in \text{cl}(S)$, we also have $\mathbf{p}_{k_j}^T \bar{\mathbf{x}} \leq \alpha_{k_j}$. Thus

$$\mathbf{p}_{k_j}^T (\mathbf{x} - \bar{\mathbf{x}}) = \mathbf{p}_{k_j}^T \mathbf{x} - \mathbf{p}_{k_j}^T \bar{\mathbf{x}} \leq \alpha_{k_j} - \mathbf{p}_{k_j}^T \bar{\mathbf{x}}.$$

Since $\mathbf{p}_{k_j}^T \mathbf{y}_{k_j} > \alpha_{k_j} \geq \mathbf{p}_{k_j}^T \bar{\mathbf{x}}$ and $\mathbf{y}_{k_j} \rightarrow \bar{\mathbf{x}}$, taking the limit gives $\mathbf{p}^T \bar{\mathbf{x}} \geq \mathbf{p}^T \bar{\mathbf{x}}$, which is consistent. More directly, for any $\mathbf{x} \in \text{cl}(S)$, taking $j \rightarrow \infty$ in the inequality $\mathbf{p}_{k_j}^T \mathbf{x} \leq \mathbf{p}_{k_j}^T \mathbf{y}_{k_j}$ yields $\mathbf{p}^T \mathbf{x} \leq \mathbf{p}^T \bar{\mathbf{x}}$, i.e., $\mathbf{p}^T (\mathbf{x} - \bar{\mathbf{x}}) \leq 0$. \square

Notice: Draft Material Ahead

The material from this point forward is subject to modification and will be verified as we progress through the course. While the content is based on standard optimization theory and the course textbooks, please:

- Cross-reference important results with the textbooks
- Note any discrepancies or unclear points to discuss in class
- Check back periodically for updated versions of these notes

Material will be “promoted” to verified status as we cover it in lectures.

Part II

Convex Analysis

Chapter 4

Convex Functions and Subgradients

This chapter develops the theory of convex and concave functions, which plays a central role in optimization. We begin with the definitions of convex, concave, strictly convex, and strictly concave functions, along with illustrative examples. We then examine operations that preserve convexity, which allow us to construct complex convex functions from simpler building blocks. The continuity properties of convex functions and the relationship between convex functions and convex sets through epigraphs provide important geometric insights. For differentiable functions, we present the powerful first-order and second-order characterizations of convexity, which connect the analytical properties of convexity to gradient and Hessian conditions. The chapter culminates with the theory of subgradients, which generalizes the concept of derivatives to nondifferentiable convex functions, and establishes the fundamental optimality conditions for convex optimization problems.

Recommended Reading

- Sections 3.1, 3.2, and 3.3 of Bazaraa, Sherali, and Shetty (2006) — definitions, operations, and differentiable convex functions
- Sections 3.1 and 3.2 of Boyd and Vandenberghe — convex functions and first/second-order conditions
- **Supplementary:** Sections 2.1–2.3 of Wright and Recht (2022) — convexity from a data science perspective

4.1 Convex and Concave Functions

Let $X \subseteq \mathbb{R}^n$ be a convex set. The following definitions establish the fundamental concepts of convexity and concavity for functions.

Definition 4.1 (Convex Function). A function $f : X \rightarrow \mathbb{R}$ is said to be **convex** if for *any* two points \mathbf{x}_1 and \mathbf{x}_2 in X and *any* $\lambda \in [0, 1]$,

$$f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda)f(\mathbf{x}_2).$$

Definition 4.2 (Concave Function). A function $f : X \rightarrow \mathbb{R}$ is said to be **concave** if $-f$ is convex, i.e., for *any* two points \mathbf{x}_1 and \mathbf{x}_2 in X and *any* $\lambda \in [0, 1]$,

$$f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \geq \lambda f(\mathbf{x}_1) + (1 - \lambda)f(\mathbf{x}_2).$$

The geometric interpretation of convexity is that the line segment connecting any two points on the graph of the function lies above (or on) the graph itself. For concave functions, the line segment lies below (or on) the graph.

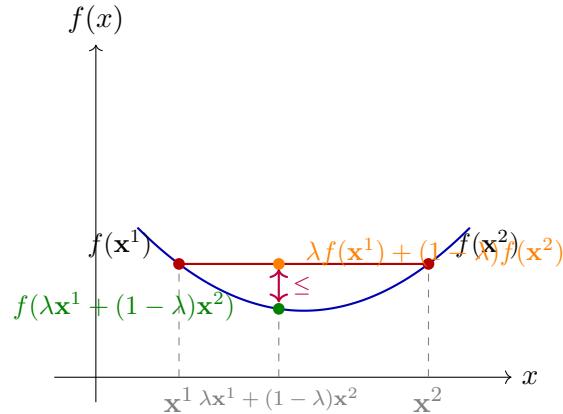


Figure 4.1: Geometric interpretation of convexity: the chord (red line) lies above the graph of the function (blue curve). The function value at any convex combination (green point) is at most the corresponding convex combination of function values (orange point).

Remark 4.1. Affine functions of the form $\mathbf{a}^T \mathbf{x} + b$ are *both* convex and concave. In fact, affine functions are the *only* functions that are simultaneously convex and concave.

4.1.1 Strictly Convex and Strictly Concave Functions

Strict convexity and strict concavity impose stronger requirements on the function's curvature.

Definition 4.3 (Strictly Convex Function). A function $f : X \rightarrow \mathbb{R}$ is said to be **strictly convex** if for *any* two *distinct* points \mathbf{x}_1 and \mathbf{x}_2 in X and *any* $\lambda \in (0, 1)$,

$$f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) < \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2).$$

Definition 4.4 (Strictly Concave Function). A function $f : X \rightarrow \mathbb{R}$ is said to be **strictly concave** if $-f$ is strictly convex, i.e., for *any* two *distinct* points \mathbf{x}_1 and \mathbf{x}_2 in X and *any* $\lambda \in (0, 1)$,

$$f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) > \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2).$$

Note that the strict inequality and the requirement that $\lambda \in (0, 1)$ (open interval) rather than $\lambda \in [0, 1]$ (closed interval) distinguish strict convexity from ordinary convexity. The exclusion of the endpoints ensures that we are considering genuine convex combinations rather than trivial cases.

Remark 4.2. Affine functions are *not* strictly convex or strictly concave. This follows because for any affine function $f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} + b$, we have equality:

$$f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) = \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2).$$

4.1.2 Examples of Convex and Concave Functions

The following examples illustrate the variety of convex and concave functions encountered in optimization.

Example 4.1 (Convex Functions). The following are examples of convex functions:

1. **Exponential function:** $f(x) = \exp(x)$ on $X = \mathbb{R}$. This function is strictly convex.
2. **Power functions:** $f(x) = x^p$ on $X = (0, +\infty)$ for $p \geq 1$ or $p \leq 0$. When $p > 1$ or $p < 0$, the function is strictly convex.
3. **Negative entropy:** $f(x) = x \ln(x)$ on $X = (0, +\infty)$. This function is strictly convex.
4. **Quadratic functions:** $f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$ on $X = \mathbb{R}^n$ for any positive semidefinite matrix $A \in \mathcal{S}_+^n$, $\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$. The function is strictly convex if A is positive definite.

Example 4.2 (Concave Functions). The following are examples of concave functions:

1. **Power functions:** $f(x) = x^p$ on $X = (0, +\infty)$ for $p \in [0, 1]$. When $p \in (0, 1)$, the function is strictly concave.
2. **Geometric mean:** $f(\mathbf{x}) = \left(\prod_{i=1}^n x_i \right)^{1/n}$ on $X = \mathbb{R}_{++}^n$ (the positive orthant). This function is concave.
3. **Logarithm:** $f(x) = \ln(x)$ on $X = (0, +\infty)$. This function is strictly concave.
4. **Quadratic functions:** $f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$ on $X = \mathbb{R}^n$ for any negative semidefinite matrix A (i.e., $-A \in \mathcal{S}_+^n$), $\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$. The function is strictly concave if A is negative definite.

4.2 Operations Preserving Convexity

One of the most useful aspects of convex functions is that convexity is preserved under various operations. This allows us to construct and verify convexity of complex functions by combining simpler convex functions.

Theorem 4.1 (Operations Preserving Convexity). *Suppose $f, f_1, \dots, f_m : \mathbb{R}^n \rightarrow \mathbb{R}$ are convex functions. Then the following functions are also convex:*

1. **Pointwise Sum:** $g(\mathbf{x}) := f_1(\mathbf{x}) + f_2(\mathbf{x})$.
2. **Nonnegative Scalar Multiplication:** $g(\mathbf{x}) := \alpha f(\mathbf{x})$ for any $\alpha \geq 0$.
3. **Pointwise Maximum:** $g(\mathbf{x}) := \max_{i \in \{1, \dots, m\}} f_i(\mathbf{x})$.
4. **Affine Composition:** $g(\mathbf{x}) := f(A\mathbf{x} + \mathbf{b})$ for any $A \in \mathbb{R}^{k \times n}$, $\mathbf{b} \in \mathbb{R}^k$.
5. **General Composition:** $h(\mathbf{x}) := f(g(\mathbf{x}))$, where $f : \mathbb{R} \rightarrow \mathbb{R}$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}$, whenever either:
 - (i) g is convex and f is both convex and nondecreasing, or
 - (ii) g is concave and f is both convex and nonincreasing.

Proof. We prove parts (1) and (3) as representative cases.

Part (1): Let $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and $\lambda \in [0, 1]$. Then

$$\begin{aligned} g(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) &= f_1(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) + f_2(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \\ &\leq \lambda f_1(\mathbf{x}_1) + (1 - \lambda)f_1(\mathbf{x}_2) + \lambda f_2(\mathbf{x}_1) + (1 - \lambda)f_2(\mathbf{x}_2) \\ &= \lambda[f_1(\mathbf{x}_1) + f_2(\mathbf{x}_1)] + (1 - \lambda)[f_1(\mathbf{x}_2) + f_2(\mathbf{x}_2)] \\ &= \lambda g(\mathbf{x}_1) + (1 - \lambda)g(\mathbf{x}_2). \end{aligned}$$

Part (3): Let $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and $\lambda \in [0, 1]$. For each i ,

$$f_i(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \leq \lambda f_i(\mathbf{x}_1) + (1 - \lambda)f_i(\mathbf{x}_2) \leq \lambda g(\mathbf{x}_1) + (1 - \lambda)g(\mathbf{x}_2),$$

where the last inequality follows because $f_i(\mathbf{x}_j) \leq g(\mathbf{x}_j)$ for all i and j . Taking the maximum over i on the left side gives the result. \square

Example 4.3 (Convexity of Optimal Value Functions). Consider the function

$$f(\mathbf{x}) := \min\{\mathbf{g}^T \mathbf{y} : A\mathbf{y} \geq \mathbf{b} - G\mathbf{x}, \mathbf{y} \in \mathbb{R}^p\}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{g} \in \mathbb{R}^p$, $A \in \mathbb{R}^{m \times p}$, $\mathbf{b} \in \mathbb{R}^m$, and $G \in \mathbb{R}^{m \times n}$.

Assume that for each $\mathbf{x} \in \mathbb{R}^n$, there exists an optimal solution to the

above linear program. Is the function f convex on \mathbb{R}^n ?

Solution: Yes, f is convex. To see this, note that by linear programming duality, we can write

$$f(\mathbf{x}) = \max\{(\mathbf{b} - G\mathbf{x})^T \boldsymbol{\pi} : A^T \boldsymbol{\pi} = \mathbf{g}, \boldsymbol{\pi} \geq \mathbf{0}\}.$$

For each feasible $\boldsymbol{\pi}$, the function $(\mathbf{b} - G\mathbf{x})^T \boldsymbol{\pi} = \mathbf{b}^T \boldsymbol{\pi} - \boldsymbol{\pi}^T G\mathbf{x}$ is affine (hence convex) in \mathbf{x} . Since $f(\mathbf{x})$ is the pointwise maximum over a family of affine functions, it is convex by Theorem 4.1(3).

4.3 Continuity of Convex Functions

Convex functions possess remarkable regularity properties. In particular, they are automatically continuous on the interior of their domain.

Theorem 4.2 (Continuity of Convex Functions). *Let $S \subseteq \mathbb{R}^n$ be a nonempty convex set, and let $f : S \rightarrow \mathbb{R}$ be a convex function. Then f is continuous on the interior of S .*

The proof is a reading assignment (see Theorem 3.1.3 of Bazaraa et al.).

Remark 4.3. The theorem states that discontinuities of a convex function can only occur at boundary points of its domain. This is a significant structural property that simplifies the analysis of convex functions.

Example 4.4 (Discontinuous Convex Function on Closed Domain).

Consider $S := \{x \in \mathbb{R} : -1 \leq x \leq 1\}$ and

$$f(x) = \begin{cases} x^2 & \text{if } |x| < 1, \\ 2 & \text{if } |x| = 1. \end{cases}$$

The function f is convex on S but discontinuous at the boundary points $x = \pm 1$. This does not contradict the theorem since ± 1 are not interior points of S .

Example 4.5 (Discontinuous Convex Function in Higher Dimensions).

Consider $S := \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\|_2 \leq 1\}$ and

$$f(\mathbf{x}) = \begin{cases} 0 & \text{if } \|\mathbf{x}\|_2 < 1, \\ 1 & \text{if } \|\mathbf{x}\|_2 = 1. \end{cases}$$

The function f is convex on S but discontinuous at every boundary point. Again, this is consistent with the theorem since all points of discontinuity lie on the boundary of S .

4.4 Characterizations of Differentiable Convex Functions

For differentiable convex functions, we have powerful characterizations using derivatives. Before presenting these results, we briefly review the necessary calculus background.

4.4.1 Calculus Background

This subsection collects the calculus prerequisites needed for the characterizations of differentiable convex functions. We introduce first-order concepts (derivatives, gradients) and second-order concepts (Hessians, positive semidefiniteness), then present Taylor's theorem which provides the key tool for connecting these concepts to convexity.

First-Order Concepts

Definition 4.5 (Derivative). For $f : \mathbb{R} \rightarrow \mathbb{R}$, the **derivative** at x is

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h},$$

provided this limit exists.

Definition 4.6 (Partial Derivative). For $f : \mathbb{R}^n \rightarrow \mathbb{R}$, the **partial**

derivative with respect to x_i at \mathbf{x} is

$$\frac{\partial f}{\partial x_i}(\mathbf{x}) = \lim_{h \rightarrow 0} \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x})}{h},$$

where \mathbf{e}_i is the i -th standard basis vector.

Definition 4.7 (Gradient). For $f : \mathbb{R}^n \rightarrow \mathbb{R}$ differentiable, the **gradient** is the vector of all partial derivatives:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^n.$$

The gradient $\nabla f(\mathbf{x})$ points in the direction of steepest increase of f at \mathbf{x} , and $\|\nabla f(\mathbf{x})\|$ gives the rate of increase in that direction. Critical points, where $\nabla f(\mathbf{x}) = \mathbf{0}$, are candidates for local minima, maxima, or saddle points.

Definition 4.8 (Directional Derivative). For $f : \mathbb{R}^n \rightarrow \mathbb{R}$ differentiable at \mathbf{x} , the **directional derivative** in direction $\mathbf{d} \in \mathbb{R}^n$ is

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

Second-Order Concepts

Definition 4.9 (Hessian Matrix). For $f : \mathbb{R}^n \rightarrow \mathbb{R}$ twice differentiable, the **Hessian** is the matrix of second partial derivatives:

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

For smooth functions (when mixed partials are continuous), the Hessian is symmetric.

Example 4.6 (Running Example: Gradient and Hessian). Consider $f(x, y) = x^2 + 2y^2$. The partial derivatives are

$$\frac{\partial f}{\partial x} = 2x, \quad \frac{\partial f}{\partial y} = 4y,$$

so the gradient is $\nabla f(x, y) = \begin{pmatrix} 2x \\ 4y \end{pmatrix}$. For example, at $(1, 2)$: $\nabla f(1, 2) = \begin{pmatrix} 2 \\ 8 \end{pmatrix}$.

The second partial derivatives are

$$\frac{\partial^2 f}{\partial x^2} = 2, \quad \frac{\partial^2 f}{\partial y^2} = 4, \quad \frac{\partial^2 f}{\partial x \partial y} = 0,$$

so the Hessian is $\nabla^2 f(x, y) = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$. Note that the Hessian is constant (independent of x and y) for this quadratic function.

Definition 4.10 (Positive Semidefinite and Positive Definite Matrices). A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is:

- **Positive semidefinite** ($A \succeq 0$) if $\mathbf{d}^T A \mathbf{d} \geq 0$ for all $\mathbf{d} \in \mathbb{R}^n$.
- **Positive definite** ($A \succ 0$) if $\mathbf{d}^T A \mathbf{d} > 0$ for all $\mathbf{d} \neq \mathbf{0}$.

Theorem 4.3 (Equivalent Conditions for Positive (Semi)Definiteness).

For a symmetric matrix $A \in \mathbb{R}^{n \times n}$, the following are equivalent:

1. $A \succeq 0$ (respectively, $A \succ 0$).
2. All eigenvalues of A are ≥ 0 (respectively, > 0).
3. All leading principal minors are ≥ 0 (respectively, > 0 for positive definiteness).
4. $A = B^T B$ for some matrix B (respectively, B has full column rank).

Proof. We prove the equivalences for the positive semidefinite case; the positive definite case follows by replacing weak inequalities with strict ones throughout.

(1) \Leftrightarrow (2): Since A is symmetric, the Spectral Theorem guarantees that A has an orthonormal basis of eigenvectors. Write $A = Q\Lambda Q^T$, where Q is orthogonal and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ contains the eigenvalues. For any $\mathbf{d} \in \mathbb{R}^n$, let $\mathbf{z} = Q^T \mathbf{d}$. Then

$$\mathbf{d}^T A \mathbf{d} = \mathbf{d}^T Q \Lambda Q^T \mathbf{d} = \mathbf{z}^T \Lambda \mathbf{z} = \sum_{i=1}^n \lambda_i z_i^2.$$

If all $\lambda_i \geq 0$, then $\mathbf{d}^T A \mathbf{d} \geq 0$ for all \mathbf{d} , so $A \succeq 0$. Conversely, if $A \succeq 0$, taking $\mathbf{d} = \mathbf{q}_i$ (the i -th column of Q) gives $\lambda_i = \mathbf{q}_i^T A \mathbf{q}_i \geq 0$.

(1) \Rightarrow (4): The Spectral Theorem gives $A = Q\Lambda Q^T$ with $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ and $\lambda_i \geq 0$. Define $\Lambda^{1/2} = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$ and set $B = \Lambda^{1/2} Q^T$. Then $B^T B = Q \Lambda^{1/2} \Lambda^{1/2} Q^T = Q \Lambda Q^T = A$.

(4) \Rightarrow (1): If $A = B^T B$, then for any $\mathbf{d} \in \mathbb{R}^n$,

$$\mathbf{d}^T A \mathbf{d} = \mathbf{d}^T B^T B \mathbf{d} = \|B \mathbf{d}\|^2 \geq 0.$$

(2) \Rightarrow (3) for positive definite: For the positive definite case, this follows from the fact that the determinant of a matrix equals the product of its eigenvalues. Each leading principal submatrix of a positive definite matrix is itself positive definite (by restricting the quadratic form), hence has positive eigenvalues and thus positive determinant.

(3) \Rightarrow (1) for positive definite: This is known as Sylvester's criterion. The proof proceeds by induction on n . For $n = 1$, $A = (a_{11})$ with $a_{11} > 0$ is clearly positive definite. For the inductive step, one shows that if all

leading principal minors are positive, the matrix can be reduced via Gaussian elimination (which preserves positive definiteness) to a form where the result follows from the inductive hypothesis.

Remark 4.4. For positive semidefiniteness, condition (3) requires checking *all* principal minors (not just leading ones), and they must all be ≥ 0 . This is because a matrix like $\begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}$ has nonnegative leading principal minors (0 and 0) but is not positive semidefinite.

□

Example 4.7 (Checking Positive Definiteness). For the Hessian $\nabla^2 f = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$ from Example 4.6, the eigenvalues are 2 and 4, both strictly positive. Therefore, $\nabla^2 f \succ 0$ (positive definite).

Taylor's Theorem

Taylor's theorem provides polynomial approximations to smooth functions and is fundamental to the analysis of convexity and optimization algorithms.

Theorem 4.4 (Taylor's Theorem – Univariate). *Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be k times continuously differentiable on an open interval containing x and $x + h$. Then*

$$f(x + h) = f(x) + f'(x)h + \frac{f''(x)}{2!}h^2 + \cdots + \frac{f^{(k-1)}(x)}{(k-1)!}h^{k-1} + R_k,$$

where the remainder R_k can be expressed in several equivalent forms:

1. **Lagrange form:** There exists ξ strictly between x and $x + h$ such that

$$R_k = \frac{f^{(k)}(\xi)}{k!}h^k.$$

2. **Integral form:**

$$R_k = \frac{1}{(k-1)!} \int_x^{x+h} (x + h - t)^{k-1} f^{(k)}(t) dt.$$

Proof. We prove the Lagrange form for $k = 1$ (the Mean Value Theorem) and sketch the general case.

For $k = 1$, we must show $f(x + h) = f(x) + f'(\xi)h$ for some ξ between x and $x + h$. Define $g(t) = f(t) - f(x) - \frac{f(x+h)-f(x)}{h}(t-x)$ on $[x, x+h]$. Then $g(x) = g(x+h) = 0$. By Rolle's theorem, there exists $\xi \in (x, x+h)$ with $g'(\xi) = 0$. Computing $g'(t) = f'(t) - \frac{f(x+h)-f(x)}{h}$, we obtain $f'(\xi) = \frac{f(x+h)-f(x)}{h}$, which gives $f(x+h) = f(x) + f'(\xi)h$.

For general k , one applies a similar argument using a carefully constructed auxiliary function and the generalized Rolle's theorem. The integral form follows from repeated integration by parts. \square

Corollary 4.5 (Mean Value Theorem). *If $f : \mathbb{R} \rightarrow \mathbb{R}$ is differentiable on an open interval containing $[a, b]$, then there exists $\xi \in (a, b)$ such that*

$$f(b) - f(a) = f'(\xi)(b - a).$$

Theorem 4.6 (Taylor's Theorem – Multivariate). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be twice continuously differentiable on an open convex set S containing \mathbf{x} and $\mathbf{x} + \mathbf{d}$. Then:*

1. **First-order expansion with exact remainder:** There exists $\theta \in (0, 1)$ such that

$$f(\mathbf{x} + \mathbf{d}) = f(\mathbf{x}) + \nabla f(\mathbf{x} + \theta\mathbf{d})^T \mathbf{d}.$$

2. **Second-order expansion with exact remainder:** There exists $\theta \in (0, 1)$ such that

$$f(\mathbf{x} + \mathbf{d}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 f(\mathbf{x} + \theta\mathbf{d}) \mathbf{d}.$$

3. **Second-order expansion with $o(\|\mathbf{d}\|^2)$ remainder:**

$$f(\mathbf{x} + \mathbf{d}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 f(\mathbf{x}) \mathbf{d} + o(\|\mathbf{d}\|^2).$$

Proof. The key idea is to reduce to the univariate case. Define the function $\phi : [0, 1] \rightarrow \mathbb{R}$ by $\phi(t) = f(\mathbf{x} + t\mathbf{d})$. By the chain rule,

$$\phi'(t) = \nabla f(\mathbf{x} + t\mathbf{d})^T \mathbf{d}, \quad \phi''(t) = \mathbf{d}^T \nabla^2 f(\mathbf{x} + t\mathbf{d}) \mathbf{d}.$$

Part 1: Applying the univariate Mean Value Theorem to ϕ on $[0, 1]$, there exists $\theta \in (0, 1)$ such that $\phi(1) - \phi(0) = \phi'(\theta)$. Substituting the definitions gives the result.

Part 2: Applying Taylor's theorem with $k = 2$ to ϕ , there exists $\theta \in (0, 1)$ such that

$$\phi(1) = \phi(0) + \phi'(0) \cdot 1 + \frac{\phi''(\theta)}{2} \cdot 1^2.$$

Substituting $\phi(1) = f(\mathbf{x} + \mathbf{d})$, $\phi(0) = f(\mathbf{x})$, $\phi'(0) = \nabla f(\mathbf{x})^T \mathbf{d}$, and $\phi''(\theta) = \mathbf{d}^T \nabla^2 f(\mathbf{x} + \theta\mathbf{d}) \mathbf{d}$ yields the result.

Part 3: This follows from part 2 and the continuity of the Hessian: as $\|\mathbf{d}\| \rightarrow 0$, the point $\mathbf{x} + \theta\mathbf{d} \rightarrow \mathbf{x}$, so $\nabla^2 f(\mathbf{x} + \theta\mathbf{d}) \rightarrow \nabla^2 f(\mathbf{x})$. \square

Remark 4.5. The exact remainder forms (parts 1 and 2) are essential for proving the first-order and second-order conditions for convexity. The point $\mathbf{x} + \theta\mathbf{d}$ lies on the line segment between \mathbf{x} and $\mathbf{x} + \mathbf{d}$, but its precise location (θ) depends on the function and the points involved.

4.4.2 First-Order Condition for Convexity

The first-order condition characterizes convexity in terms of the gradient.

Theorem 4.7 (First-Order Condition for Convexity). *Let $S \subseteq \mathbb{R}^n$ be a nonempty open convex set, and let $f : S \rightarrow \mathbb{R}$ be differentiable on S . Then f is convex on S if and only if*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) \quad \text{for all } \mathbf{x}, \mathbf{y} \in S.$$

The geometric interpretation is that the graph of a convex function lies above all of its tangent hyperplanes. The linear function $\ell(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x})$ is the first-order Taylor approximation of f at \mathbf{x} , and for convex functions, this approximation is always a global underestimator.

Proof. (\Rightarrow) Assume f is convex. For $\mathbf{x}, \mathbf{y} \in S$ and $\lambda \in (0, 1]$, by convexity:

$$f(\mathbf{x} + \lambda(\mathbf{y} - \mathbf{x})) \leq (1 - \lambda)f(\mathbf{x}) + \lambda f(\mathbf{y}).$$

Rearranging:

$$\frac{f(\mathbf{x} + \lambda(\mathbf{y} - \mathbf{x})) - f(\mathbf{x})}{\lambda} \leq f(\mathbf{y}) - f(\mathbf{x}).$$

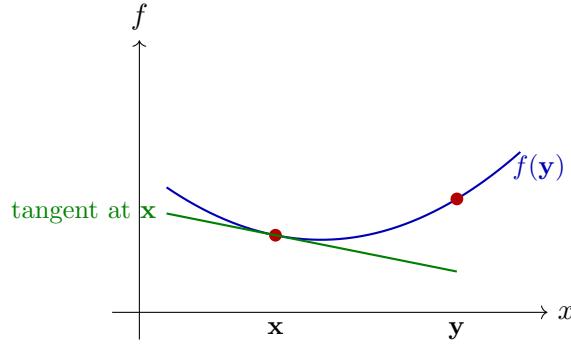


Figure 4.2: The first-order condition: the graph of a convex function lies above all its tangent lines.

Taking the limit as $\lambda \rightarrow 0^+$:

$$\nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq f(\mathbf{y}) - f(\mathbf{x}),$$

which gives the desired inequality.

(\Leftarrow) Assume the first-order condition holds. For $\mathbf{x}, \mathbf{y} \in S$ and $\lambda \in [0, 1]$, let $\mathbf{z} = \lambda\mathbf{x} + (1 - \lambda)\mathbf{y}$. Applying the first-order condition at \mathbf{z} :

$$\begin{aligned} f(\mathbf{x}) &\geq f(\mathbf{z}) + \nabla f(\mathbf{z})^T(\mathbf{x} - \mathbf{z}), \\ f(\mathbf{y}) &\geq f(\mathbf{z}) + \nabla f(\mathbf{z})^T(\mathbf{y} - \mathbf{z}). \end{aligned}$$

Multiplying the first inequality by λ and the second by $(1 - \lambda)$ and adding:

$$\begin{aligned} \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y}) &\geq f(\mathbf{z}) + \nabla f(\mathbf{z})^T[\lambda(\mathbf{x} - \mathbf{z}) + (1 - \lambda)(\mathbf{y} - \mathbf{z})] \\ &= f(\mathbf{z}) + \nabla f(\mathbf{z})^T[\lambda\mathbf{x} + (1 - \lambda)\mathbf{y} - \mathbf{z}] \\ &= f(\mathbf{z}) + \nabla f(\mathbf{z})^T\mathbf{0} = f(\mathbf{z}). \end{aligned}$$

This establishes convexity of f . □

4.4.3 Second-Order Condition for Convexity

For twice-differentiable functions, we have a characterization in terms of the Hessian.

Theorem 4.8 (Second-Order Condition for Convexity). *Let $S \subseteq \mathbb{R}^n$ be a nonempty open convex set, and let $f : S \rightarrow \mathbb{R}$ be twice continuously*

differentiable on S . Then:

1. *f is convex on S if and only if $\nabla^2 f(\mathbf{x}) \succeq 0$ for all $\mathbf{x} \in S$.*
2. *If $\nabla^2 f(\mathbf{x}) \succ 0$ for all $\mathbf{x} \in S$, then f is strictly convex on S .*

Remark 4.6. For $f : \mathbb{R} \rightarrow \mathbb{R}$, the Hessian is just the second derivative $f''(x)$. The theorem simplifies to:

- f is convex if and only if $f''(x) \geq 0$ for all x .
- If $f''(x) > 0$ for all x , then f is strictly convex.

Remark 4.7. The converse of part (2) is *not* true. For example, $f(x) = x^4$ is strictly convex on \mathbb{R} , but $f''(0) = 0$. Strict convexity does not require the Hessian to be positive definite everywhere.

Proof Sketch. (\Rightarrow for part 1) Assume f is convex. By the first-order condition (Theorem 4.7):

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \quad \text{for all } \mathbf{x}, \mathbf{y} \in S.$$

Let $\mathbf{y} = \mathbf{x} + t\mathbf{d}$ for small $t > 0$ and any direction \mathbf{d} . By Taylor expansion:

$$f(\mathbf{x} + t\mathbf{d}) = f(\mathbf{x}) + t\nabla f(\mathbf{x})^T\mathbf{d} + \frac{t^2}{2}\mathbf{d}^T\nabla^2 f(\mathbf{x})\mathbf{d} + o(t^2).$$

Substituting into the first-order condition and dividing by $\frac{t^2}{2}$:

$$\mathbf{d}^T \nabla^2 f(\mathbf{x}) \mathbf{d} \geq 0 \quad \text{for all } \mathbf{d}.$$

This means $\nabla^2 f(\mathbf{x}) \succeq 0$.

(\Leftarrow for part 1) Uses Taylor expansion with integral form of remainder. See Bazaraa et al. Theorem 3.3.8 for the complete proof. \square

4.4.4 Running Example: Three Proofs of Convexity

We illustrate the different approaches to proving convexity using the function $f(x, y) = x^2 + 2y^2$.

Example 4.8 (Proof via Composition Rules). We prove that $f(x, y) = x^2 + 2y^2$ is convex using the preservation rules from Theorem 4.1.

Step 1: The function $g(t) = t^2$ is convex on \mathbb{R} . This can be verified directly from the definition: for $t_1, t_2 \in \mathbb{R}$ and $\lambda \in [0, 1]$,

$$(\lambda t_1 + (1 - \lambda)t_2)^2 \leq \lambda t_1^2 + (1 - \lambda)t_2^2$$

follows from expanding and rearranging to get $\lambda(1 - \lambda)(t_1 - t_2)^2 \geq 0$.

Step 2: Define $g_1(x, y) = x^2$ and $g_2(x, y) = y^2$. These are compositions of the convex function t^2 with linear functions, so both are convex.

Step 3: Write $f(x, y) = 1 \cdot g_1(x, y) + 2 \cdot g_2(x, y)$. This is a nonnegative weighted sum (weights $1 \geq 0$ and $2 \geq 0$) of convex functions.

By Theorem 4.1, $f(x, y) = x^2 + 2y^2$ is convex. Moreover, since g_1 and g_2 are strictly convex and the weights are positive, f is strictly convex.

Example 4.9 (Proof via First-Order Condition). We prove that $f(x, y) = x^2 + 2y^2$ is convex using Theorem 4.7.

Step 1: Compute the gradient: $\nabla f(x, y) = \begin{pmatrix} 2x \\ 4y \end{pmatrix}$.

Step 2: The first-order condition requires: for all $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^2$,

$$f(x_2, y_2) \geq f(x_1, y_1) + \nabla f(x_1, y_1)^T \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix}.$$

Step 3: Expand both sides:

$$\begin{aligned} \text{LHS} &= x_2^2 + 2y_2^2, \\ \text{RHS} &= x_1^2 + 2y_1^2 + 2x_1(x_2 - x_1) + 4y_1(y_2 - y_1) \\ &= -x_1^2 + 2x_1x_2 - 2y_1^2 + 4y_1y_2. \end{aligned}$$

Step 4: Show LHS – RHS ≥ 0 :

$$x_2^2 + 2y_2^2 + x_1^2 - 2x_1x_2 + 2y_1^2 - 4y_1y_2 = (x_2 - x_1)^2 + 2(y_2 - y_1)^2 \geq 0.$$

This is always nonnegative, confirming convexity.

Example 4.10 (Proof via Second-Order Condition). We prove that $f(x, y) = x^2 + 2y^2$ is strictly convex using Theorem 4.8.

Step 1: Compute the gradient: $\nabla f(x, y) = \begin{pmatrix} 2x \\ 4y \end{pmatrix}$.

Step 2: Compute the Hessian: $\nabla^2 f(x, y) = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$.

Step 3: Check positive definiteness. The eigenvalues are $\lambda_1 = 2 > 0$ and $\lambda_2 = 4 > 0$.

Since $\nabla^2 f(\mathbf{x}) \succ 0$ for all \mathbf{x} , by Theorem 4.8(2), $f(x, y) = x^2 + 2y^2$ is strictly convex.

Remark 4.8 (Comparison of Methods). The three approaches to proving convexity have different strengths:

Method	Key Step	Result
Composition Rules	Sum of convex functions	Strictly convex
First-Order Condition	Show $(x_2 - x_1)^2 + 2(y_2 - y_1)^2 \geq 0$	Convex
Second-Order Condition	Eigenvalues of $\nabla^2 f$ are $2, 4 > 0$	Strictly convex

The second-order method is often the most practical for differentiable functions, as it reduces convexity checking to eigenvalue analysis of the Hessian.

4.5 Epigraph and Hypograph

The epigraph and hypograph provide a powerful connection between convex functions and convex sets, allowing us to apply results from convex set theory to the study of convex functions.

Let S be a nonempty set in \mathbb{R}^n and let $f : S \rightarrow \mathbb{R}$.

Definition 4.11 (Graph of a Function). The **graph** of f is the set

$$\text{gph}(f) := \{(\mathbf{x}, f(\mathbf{x})) : \mathbf{x} \in S\} \subset \mathbb{R}^{n+1}.$$

Definition 4.12 (Epigraph). The **epigraph** of f is the set

$$\text{epi}(f) := \{(\mathbf{x}, y) : \mathbf{x} \in S, y \in \mathbb{R}, y \geq f(\mathbf{x})\} \subset \mathbb{R}^{n+1}.$$

The epigraph consists of all points lying on or above the graph of f .

Definition 4.13 (Hypograph). The **hypograph** of f is the set

$$\text{hypo}(f) := \{(\mathbf{x}, y) : \mathbf{x} \in S, y \in \mathbb{R}, y \leq f(\mathbf{x})\} \subset \mathbb{R}^{n+1}.$$

The hypograph consists of all points lying on or below the graph of f .

4.5.1 Epigraph as the Link Between Convex Sets and Functions

The following theorem establishes the fundamental connection between convex functions and convex sets.

Theorem 4.9 (Convexity and Epigraph). *Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$. Then f is a convex function if and only if $\text{epi}(f)$ is a convex set.*

Proof. (\Rightarrow) Suppose f is convex. Let $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \in \text{epi}(f)$ and $\lambda \in [0, 1]$. Then $y_1 \geq f(\mathbf{x}_1)$ and $y_2 \geq f(\mathbf{x}_2)$. We need to show that $(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2, \lambda y_1 + (1 - \lambda)y_2) \in \text{epi}(f)$.

Since S is convex, $\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in S$. Moreover, by convexity of f :

$$\begin{aligned} f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) &\leq \lambda f(\mathbf{x}_1) + (1 - \lambda)f(\mathbf{x}_2) \\ &\leq \lambda y_1 + (1 - \lambda)y_2. \end{aligned}$$

Thus $(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2, \lambda y_1 + (1 - \lambda)y_2) \in \text{epi}(f)$.

(\Leftarrow) Suppose $\text{epi}(f)$ is convex. Let $\mathbf{x}_1, \mathbf{x}_2 \in S$ and $\lambda \in [0, 1]$. Then $(\mathbf{x}_1, f(\mathbf{x}_1)), (\mathbf{x}_2, f(\mathbf{x}_2)) \in \text{epi}(f)$. By convexity of $\text{epi}(f)$:

$$(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2, \lambda f(\mathbf{x}_1) + (1 - \lambda)f(\mathbf{x}_2)) \in \text{epi}(f).$$

This means $f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda)f(\mathbf{x}_2)$, so f is convex. \square

Corollary 4.10. *The hypograph of a concave function is a convex set.*

Remark 4.9. Since the epigraph of a convex function and the hypograph of a concave function are convex sets, they admit supporting hyperplanes at boundary points. This observation leads directly to the concept of subgradients, which we develop in the next section.

4.6 Subgradients

For differentiable convex functions, the gradient provides complete first-order information about the function. However, many important convex functions are not differentiable everywhere (e.g., $f(x) = |x|$ at $x = 0$). The concept of subgradient generalizes the gradient to handle nondifferentiable convex functions.

Definition 4.14 (Subgradient of a Convex Function). Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be a convex function. A vector $\xi \in \mathbb{R}^n$ is called a **subgradient** of f at $\bar{x} \in S$ if

$$f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \xi^T(\mathbf{x} - \bar{\mathbf{x}}) \quad \text{for all } \mathbf{x} \in S.$$

Definition 4.15 (Subgradient of a Concave Function). Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be a concave function. A vector $\xi \in \mathbb{R}^n$ is called a **subgradient** of f at $\bar{x} \in S$ if

$$f(\mathbf{x}) \leq f(\bar{\mathbf{x}}) + \xi^T(\mathbf{x} - \bar{\mathbf{x}}) \quad \text{for all } \mathbf{x} \in S.$$

Remark 4.10 (Geometric Interpretation). The inequality $f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \xi^T(\mathbf{x} - \bar{\mathbf{x}})$ states that the affine function $\ell(\mathbf{x}) = f(\bar{\mathbf{x}}) + \xi^T(\mathbf{x} - \bar{\mathbf{x}})$ is a global underestimator of f . Geometrically, the graph of ℓ defines a hyperplane in \mathbb{R}^{n+1} that:

- passes through the point $(\bar{\mathbf{x}}, f(\bar{\mathbf{x}}))$ on the graph of f ,
- supports the epigraph of f from below.

The subgradient vector ξ corresponds to the slope of this supporting hyperplane.

Definition 4.16 (Subdifferential). The set of all subgradients of f at \bar{x} is called the **subdifferential** of f at \bar{x} , denoted $\partial f(\bar{x}) \subseteq \mathbb{R}^n$.

Example 4.11 (Subdifferential of Absolute Value Function). Consider $f(x) = |x|$ on \mathbb{R} . At $\bar{x} = 0$, a vector $\xi \in \mathbb{R}$ is a subgradient if and only

if

$$|x| \geq |0| + \xi(x - 0) = \xi x \quad \text{for all } x \in \mathbb{R}.$$

This inequality holds if and only if $\xi \in [-1, 1]$. Therefore, $\partial f(0) = [-1, 1]$.

At any $\bar{x} > 0$, the function is differentiable with derivative 1, so $\partial f(\bar{x}) = \{1\}$. Similarly, for $\bar{x} < 0$, $\partial f(\bar{x}) = \{-1\}$.

4.6.1 Existence of Subgradients

A natural question is whether subgradients always exist. The following theorem provides a positive answer for interior points.

Theorem 4.11 (Existence of Subgradients). *Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be a convex function. Then for $\bar{\mathbf{x}} \in \text{int}(S)$, there exists a vector ξ such that the hyperplane*

$$H = \{(\mathbf{x}, y) : y = f(\bar{\mathbf{x}}) + \xi^T(\mathbf{x} - \bar{\mathbf{x}})\}$$

supports $\text{epi}(f)$ at $(\bar{\mathbf{x}}, f(\bar{\mathbf{x}}))$. In particular, ξ is a subgradient of f at $\bar{\mathbf{x}}$.

The proof is a reading assignment (see Theorem 3.2.5 of Bazaraa et al.). The key idea is to apply the supporting hyperplane theorem to the epigraph of f at the boundary point $(\bar{\mathbf{x}}, f(\bar{\mathbf{x}}))$.

Corollary 4.12. *A convex function has at least one subgradient at every point in the interior of its domain.*

Remark 4.11 (Properties of the Subdifferential). 1. If f is differentiable at $\bar{\mathbf{x}}$, then $\nabla f(\bar{\mathbf{x}})$ is the *unique* subgradient of f at $\bar{\mathbf{x}}$. That is, $\partial f(\bar{\mathbf{x}}) = \{\nabla f(\bar{\mathbf{x}})\}$.

2. At interior points $\bar{\mathbf{x}}$, the subdifferential $\partial f(\bar{\mathbf{x}})$ is a nonempty, compact, convex set.
3. The subdifferential provides a complete characterization of the local behavior of a convex function, even when the function is not differentiable.

The following theorem provides a converse: the existence of subgradients characterizes convexity.

Theorem 4.13 (Characterization of Convexity via Subgradients). *Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$. Suppose that for each point $\bar{\mathbf{x}} \in \text{int}(S)$, there exists a subgradient vector ξ such that*

$$f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \xi^T(\mathbf{x} - \bar{\mathbf{x}}) \quad \text{for each } \mathbf{x} \in S.$$

Then f is convex on $\text{int}(S)$.

Proof. Let $\mathbf{x}_1, \mathbf{x}_2 \in \text{int}(S)$ and $\lambda \in (0, 1)$. Define $\bar{\mathbf{x}} = \lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2$. Since $\text{int}(S)$ is convex, $\bar{\mathbf{x}} \in \text{int}(S)$.

By hypothesis, there exists a subgradient ξ of f at $\bar{\mathbf{x}}$ such that

$$\begin{aligned} f(\mathbf{x}_1) &\geq f(\bar{\mathbf{x}}) + \xi^T(\mathbf{x}_1 - \bar{\mathbf{x}}), \\ f(\mathbf{x}_2) &\geq f(\bar{\mathbf{x}}) + \xi^T(\mathbf{x}_2 - \bar{\mathbf{x}}). \end{aligned}$$

Multiplying the first inequality by λ , the second by $(1 - \lambda)$, and adding:

$$\begin{aligned} \lambda f(\mathbf{x}_1) + (1 - \lambda)f(\mathbf{x}_2) &\geq f(\bar{\mathbf{x}}) + \xi^T[\lambda(\mathbf{x}_1 - \bar{\mathbf{x}}) + (1 - \lambda)(\mathbf{x}_2 - \bar{\mathbf{x}})] \\ &= f(\bar{\mathbf{x}}) + \xi^T[\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 - \bar{\mathbf{x}}] \\ &= f(\bar{\mathbf{x}}) + \xi^T\mathbf{0} \\ &= f(\bar{\mathbf{x}}) = f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2). \end{aligned}$$

This establishes convexity of f on $\text{int}(S)$. \square

4.7 Convex Optimization

We now turn to the optimization of convex functions over convex sets. This class of problems has particularly nice properties that make them both theoretically tractable and computationally attractive.

Definition 4.17 (Convex Optimization Problem). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function and S be a nonempty convex set in \mathbb{R}^n . A **convex optimization problem** is defined by*

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in S. \end{aligned} \tag{4.1}$$

The following theorem establishes the fundamental property that makes convex optimization so attractive: local optimality implies global optimality.

Theorem 4.14 (Local Implies Global Optimality). *Suppose that $\bar{\mathbf{x}} \in S$ is a local optimal solution to Problem (4.1).*

1. *Then $\bar{\mathbf{x}}$ is a global optimal solution.*
2. *If either $\bar{\mathbf{x}}$ is a strict local minimum or f is strictly convex, then $\bar{\mathbf{x}}$ is the unique global optimal solution.*

Proof. **Part 1:** Suppose $\bar{\mathbf{x}}$ is a local minimum but not a global minimum. Then there exists $\mathbf{x}^* \in S$ with $f(\mathbf{x}^*) < f(\bar{\mathbf{x}})$. Since $\bar{\mathbf{x}}$ is a local minimum, there exists $\epsilon > 0$ such that $f(\bar{\mathbf{x}}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in S$ with $\|\mathbf{x} - \bar{\mathbf{x}}\| < \epsilon$.

Consider the point $\mathbf{x}_\lambda = \lambda\mathbf{x}^* + (1 - \lambda)\bar{\mathbf{x}}$ for small $\lambda > 0$. By convexity of S , $\mathbf{x}_\lambda \in S$. Moreover, $\|\mathbf{x}_\lambda - \bar{\mathbf{x}}\| = \lambda\|\mathbf{x}^* - \bar{\mathbf{x}}\| < \epsilon$ for sufficiently small λ .

By convexity of f :

$$f(\mathbf{x}_\lambda) \leq \lambda f(\mathbf{x}^*) + (1 - \lambda)f(\bar{\mathbf{x}}) < \lambda f(\bar{\mathbf{x}}) + (1 - \lambda)f(\bar{\mathbf{x}}) = f(\bar{\mathbf{x}}).$$

This contradicts the local optimality of $\bar{\mathbf{x}}$.

Part 2: If f is strictly convex and there exist two distinct global minima $\bar{\mathbf{x}}$ and \mathbf{x}^* , then for any $\lambda \in (0, 1)$:

$$f(\lambda\bar{\mathbf{x}} + (1 - \lambda)\mathbf{x}^*) < \lambda f(\bar{\mathbf{x}}) + (1 - \lambda)f(\mathbf{x}^*) = f(\bar{\mathbf{x}}),$$

contradicting the optimality of $\bar{\mathbf{x}}$. \square

4.7.1 Optimality Conditions

The subgradient provides necessary and sufficient conditions for optimality in convex optimization.

Theorem 4.15 (Optimality Conditions for Convex Optimization).

Consider Problem (4.1). A point $\bar{\mathbf{x}} \in S$ is an optimal solution if and only if f has a subgradient ξ at $\bar{\mathbf{x}}$ such that

$$\xi^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0 \quad \text{for all } \mathbf{x} \in S.$$

Proof. (\Rightarrow) Suppose $\bar{\mathbf{x}}$ is optimal. If $\bar{\mathbf{x}} \in \text{int}(S)$, then by Theorem 4.11, there exists a subgradient ξ at $\bar{\mathbf{x}}$. For any $\mathbf{x} \in S$, we have $f(\mathbf{x}) \geq f(\bar{\mathbf{x}})$ (by

optimality) and $f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \boldsymbol{\xi}^T(\mathbf{x} - \bar{\mathbf{x}})$ (by the subgradient inequality). The condition $\boldsymbol{\xi}^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0$ follows from the optimality condition $f(\mathbf{x}) \geq f(\bar{\mathbf{x}})$ combined with the subgradient inequality.

(\Leftarrow) Suppose there exists a subgradient $\boldsymbol{\xi}$ at $\bar{\mathbf{x}}$ with $\boldsymbol{\xi}^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0$ for all $\mathbf{x} \in S$. Then for any $\mathbf{x} \in S$:

$$f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \boldsymbol{\xi}^T(\mathbf{x} - \bar{\mathbf{x}}) \geq f(\bar{\mathbf{x}}).$$

Thus $\bar{\mathbf{x}}$ is a global minimum. \square

Corollary 4.16 (Unconstrained Optimality). *Under the assumptions of Theorem 4.15, if S is an open set (for example, $S = \mathbb{R}^n$), then $\bar{\mathbf{x}}$ is an optimal solution if and only if there exists a zero subgradient of f at $\bar{\mathbf{x}}$, i.e., $\mathbf{0} \in \partial f(\bar{\mathbf{x}})$.*

Proof. If S is open and $\boldsymbol{\xi}^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0$ for all $\mathbf{x} \in S$, then taking $\mathbf{x} = \bar{\mathbf{x}} - \epsilon\boldsymbol{\xi}$ for small $\epsilon > 0$ gives $-\epsilon\|\boldsymbol{\xi}\|^2 \geq 0$, which implies $\boldsymbol{\xi} = \mathbf{0}$. \square

Corollary 4.17 (Differentiable Case). *Assume that f is differentiable on S . Then $\bar{\mathbf{x}}$ is an optimal solution to Problem (4.1) if and only if*

$$\nabla f(\bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0 \quad \text{for all } \mathbf{x} \in S.$$

Furthermore, if S is open, then $\bar{\mathbf{x}}$ is an optimal solution if and only if $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$.

4.7.2 Applications of Optimality Conditions

Example 4.12 (Unconstrained Quadratic Minimization). Consider the problem

$$\min_{(x_1, x_2) \in \mathbb{R}^2} x_1 + x_1 x_2 + x_2^2.$$

The objective function can be written as $f(\mathbf{x}) = x_1 + x_1 x_2 + x_2^2$. To check convexity, we compute the Hessian:

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}.$$

The eigenvalues are $1 \pm \sqrt{2}$, so the Hessian is indefinite. Therefore, f is neither convex nor concave, and the problem is not a convex op-

timization problem. The standard first-order conditions $\nabla f(\mathbf{x}) = \mathbf{0}$ give:

$$\begin{cases} 1 + x_2 = 0 \\ x_1 + 2x_2 = 0 \end{cases} \implies (x_1, x_2) = (2, -1).$$

However, since the Hessian is indefinite, this is a saddle point, not a minimum.

Example 4.13 (Convex Quadratic Minimization). Consider the problem

$$\min_{(x_1, x_2) \in \mathbb{R}^2} x_1 + x_1^2 + x_2^2.$$

The Hessian is

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = 2I,$$

which is positive definite. Thus f is strictly convex, and by Corollary 4.16, the unique global minimum satisfies $\nabla f(\mathbf{x}) = \mathbf{0}$:

$$\begin{cases} 1 + 2x_1 = 0 \\ 2x_2 = 0 \end{cases} \implies (x_1, x_2) = \left(-\frac{1}{2}, 0\right).$$

Example 4.14 (Constrained Problem with No Interior Solution). Consider the problem

$$\min\{x^2 : x \in [3, 4]\}.$$

Here $f(x) = x^2$ is strictly convex and $S = [3, 4]$ is a closed convex set. The unconstrained minimizer $x = 0$ does not lie in S . At the boundary point $\bar{x} = 3$, the gradient is $\nabla f(3) = 6 > 0$. For any $x \in [3, 4]$:

$$\nabla f(3)(x - 3) = 6(x - 3) \geq 0,$$

confirming that $\bar{x} = 3$ is optimal by Corollary 4.17.

Example 4.15 (Problem with No Optimal Solution). Consider the problem

$$\min\{x^2 : x \in (3, 4)\}.$$

The feasible set $S = (3, 4)$ is open. Since $f(x) = x^2$ is strictly increasing

on $(3, 4)$, the infimum is $\inf_{x \in (3,4)} x^2 = 9$. However, this infimum is not attained since $x = 3 \notin S$. The problem has no optimal solution.

4.7.3 Existence of Optimal Solutions

The previous examples show that optimal solutions may not always exist. The following classical theorem provides sufficient conditions for existence.

Theorem 4.18 (Weierstrass' Theorem). *Let S be a nonempty, compact (closed and bounded) set, and let $f : S \rightarrow \mathbb{R}$ be continuous on S . Then the problem $\min\{f(\mathbf{x}) : \mathbf{x} \in S\}$ attains its minimum.*

Remark 4.12. The conditions in Weierstrass' theorem are sufficient but not necessary. When these conditions fail, more careful analysis is required to determine whether optimal solutions exist.

4.7.4 Ordinary Least Squares Regression

As an important application, we consider the ordinary least squares problem.

Example 4.16 (Ordinary Least Squares). Given $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, solve

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}\|_2^2.$$

Solution: The objective function is $f(\mathbf{x}) = (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}) = \mathbf{x}^T A^T A \mathbf{x} - 2\mathbf{b}^T A \mathbf{x} + \mathbf{b}^T \mathbf{b}$. This is a convex quadratic function (since $A^T A$ is positive semidefinite). The gradient is

$$\nabla f(\mathbf{x}) = 2A^T A \mathbf{x} - 2A^T \mathbf{b}.$$

Setting $\nabla f(\mathbf{x}) = \mathbf{0}$ gives the *normal equations*:

$$A^T A \mathbf{x} = A^T \mathbf{b}.$$

If $A^T A$ is invertible (which occurs when A has full column rank), the unique solution is

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

If $A^T A$ is singular, the set of optimal solutions is the affine subspace $\{\mathbf{x} : A^T A \mathbf{x} = A^T \mathbf{b}\}$.

Chapter 5

Subgradients and Optimality Conditions

This chapter develops the theory of subgradients for convex functions and establishes the fundamental optimality conditions for convex optimization problems. We begin by proving the existence of subgradients at interior points of the domain, then formulate the general convex optimization problem. The main results provide necessary and sufficient conditions for global optimality in terms of subgradients and gradients. We extend the theory to differentiable and twice differentiable convex functions, characterizing convexity through gradient and Hessian conditions. The chapter concludes with important examples including ordinary least squares regression, illustrating how these theoretical tools are applied in practice.

Recommended Reading

- Sections 2.4, 3.2, and 3.4 of Bazaraa, Sherali, and Shetty (2006)
- Sections 2.5, 3.1, and 3.2 of Boyd and Vandenberghe
- **Supplementary:** Chapter 8 of Wright and Recht (2022) — subgradients and nonsmooth optimization in ML

5.1 Existence of Subgradients of Convex Functions

A fundamental result in convex analysis is that convex functions possess subgradients at every interior point of their domain. This is established through the connection between subgradients and supporting hyperplanes

of the epigraph.

Theorem 5.1 (Existence of Subgradients). *Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be a convex function. Then for $\bar{\mathbf{x}} \in \text{int}(S)$, there exists a vector ξ such that the hyperplane*

$$H = \{(\mathbf{x}, y) : y = f(\bar{\mathbf{x}}) + \xi^T(\mathbf{x} - \bar{\mathbf{x}})\}$$

supports $\text{epi}(f)$ at $[\bar{\mathbf{x}}, f(\bar{\mathbf{x}})]$. In particular, ξ is a subgradient of f at $\bar{\mathbf{x}}$.

This theorem establishes the important fact that a convex function has a subgradient at every point in the interior of its domain.

Remark 5.1. Several important properties follow from this theorem:

- If f is differentiable at $\bar{\mathbf{x}}$, then $\nabla f(\bar{\mathbf{x}})$ is the *only* subgradient of f at $\bar{\mathbf{x}}$.
- The set of all subgradients of f at $\bar{\mathbf{x}}$ is called the **subdifferential**, denoted $\partial f(\bar{\mathbf{x}}) \subseteq \mathbb{R}^n$.
- The subdifferential $\partial f(\bar{\mathbf{x}})$ is a compact convex set at interior points $\bar{\mathbf{x}}$.

5.2 Convex Optimization Problems

We now formulate the general convex optimization problem and establish fundamental properties of its solutions.

Definition 5.1 (Convex Optimization Problem). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function and S be a nonempty convex set in \mathbb{R}^n . A **convex optimization problem** is defined by

$$\begin{aligned} & \min && f(\mathbf{x}) \\ & \text{s.t.} && \mathbf{x} \in S. \end{aligned} \tag{5.1}$$

The following theorem establishes the fundamental property that local optima are global optima in convex optimization.

Theorem 5.2 (Local Optima are Global Optima). *Suppose that $\bar{\mathbf{x}} \in S$ is a local optimal solution to Problem (5.1).*

1. *Then $\bar{\mathbf{x}}$ is a global optimal solution.*
2. *If either $\bar{\mathbf{x}}$ is a strict local minimum or f is strictly convex, then $\bar{\mathbf{x}}$ is the unique global optimal solution (it is also a strong local minimum).*

5.3 Necessary and Sufficient Conditions for a Global Minimum

The following theorem provides the fundamental characterization of optimal solutions to convex optimization problems in terms of subgradients.

Theorem 5.3 (Optimality Conditions via Subgradients). *Point $\bar{\mathbf{x}} \in S$ is an optimal solution to Problem (5.1) if and only if f has a subgradient ξ at $\bar{\mathbf{x}}$ such that*

$$\xi^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0 \quad \text{for all } \mathbf{x} \in S.$$

This theorem has important corollaries for special cases.

Corollary 5.4 (Unconstrained Optimization). *Under the assumptions of Theorem 5.3, if S is an open set (for example, $S = \mathbb{R}^n$), then $\bar{\mathbf{x}}$ is an optimal solution to the problem if and only if there exists a zero subgradient of f at $\bar{\mathbf{x}}$, i.e., $\mathbf{0} \in \partial f(\bar{\mathbf{x}})$.*

Corollary 5.5 (Differentiable Objective). *Assume that f is differentiable. Then $\bar{\mathbf{x}}$ is an optimal solution if and only if*

$$\nabla f(\bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0 \quad \text{for all } \mathbf{x} \in S.$$

Furthermore, if S is open, $\bar{\mathbf{x}}$ is an optimal solution if and only if $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$.

5.4 Applying the Necessary and Sufficient Optimality Conditions

We now illustrate the application of these optimality conditions through several examples.

Example 5.1 (Quadratic Optimization I). Consider the problem

$$\min \quad x_1 + x_1 x_2 + x_2^2 \quad \text{s.t.} \quad (x_1, x_2) \in \mathbb{R}^2.$$

To apply the optimality conditions, we first compute the gradient:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} 1 + x_2 \\ x_1 + 2x_2 \end{pmatrix}.$$

For an unconstrained problem over an open set, we need $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$:

$$\begin{cases} 1 + x_2 = 0 \\ x_1 + 2x_2 = 0 \end{cases}$$

Solving this system gives $x_2 = -1$ and $x_1 = 2$. However, we must verify that the function is convex. The Hessian is:

$$H = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}.$$

Checking positive semidefiniteness: $a = 0 \geq 0$, $c = 2 \geq 0$, but $ac - b^2 = 0 - 1 = -1 < 0$. Therefore, H is *indefinite*, the function is not convex, and we cannot apply the convex optimization theory directly.

Example 5.2 (Quadratic Optimization II). Consider the problem

$$\min \quad x_1 + x_1^2 + x_2^2 \quad \text{s.t.} \quad (x_1, x_2) \in \mathbb{R}^2.$$

The gradient is:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} 1 + 2x_1 \\ 2x_2 \end{pmatrix}.$$

Setting $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$:

$$\begin{cases} 1 + 2x_1 = 0 \\ 2x_2 = 0 \end{cases}$$

This gives $x_1 = -\frac{1}{2}$ and $x_2 = 0$. The Hessian is:

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = 2I.$$

Since H is positive definite (all eigenvalues equal 2), the function f is strictly convex. Therefore, $\bar{\mathbf{x}} = (-\frac{1}{2}, 0)$ is the unique global minimum with optimal value $f(\bar{\mathbf{x}}) = -\frac{1}{2} + \frac{1}{4} + 0 = -\frac{1}{4}$.

Example 5.3 (Constrained Optimization on a Closed Set). Consider the problem $\min\{x^2 : x \in [3, 4]\}$.

Here $f(x) = x^2$ is a convex function and $S = [3, 4]$ is a closed (compact) set. The gradient $f'(x) = 2x$ is never zero on $[3, 4]$. However, by Corollary 5.5, we need $\nabla f(\bar{x})^T(x - \bar{x}) \geq 0$ for all $x \in S$.

At $\bar{x} = 3$: $2(3)(x - 3) = 6(x - 3) \geq 0$ for all $x \in [3, 4]$. This is satisfied! Therefore, $\bar{x} = 3$ is optimal with $f(\bar{x}) = 9$.

Example 5.4 (Optimization on an Open Set). Consider the problem $\min\{x^2 : x \in (3, 4)\}$.

Here $f(x) = x^2$ is convex and $S = (3, 4)$ is an open set. For optimality on an open set, we need $f'(\bar{x}) = 2\bar{x} = 0$, which gives $\bar{x} = 0 \notin (3, 4)$.

Therefore, there is no point $\bar{x} \in S$ satisfying the optimality condition. The infimum is $\inf_{x \in (3, 4)} x^2 = 9$, but this value is not attained.

5.5 Existence of Optimal Solutions

The previous examples show that optimal solutions may not always exist. The following theorem provides sufficient conditions for existence.

Theorem 5.6 (Weierstrass' Theorem). *Let S be a nonempty, compact (closed and bounded) set, and let $f : S \rightarrow \mathbb{R}$ be continuous on S . Then the problem $\min\{f(\mathbf{x}) : \mathbf{x} \in S\}$ attains its minimum.*

Remark 5.2. The conditions in Weierstrass' theorem are sufficient but not necessary. The theorem can fail when:

- S is not closed (as in Example 5.4).
- S is not bounded.
- f is not continuous.

5.6 Ordinary Least Squares Regression

A fundamental application of convex optimization is the ordinary least squares (OLS) regression problem.

Example 5.5 (Ordinary Least Squares). Given $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, solve

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}\|_2^2.$$

Let $f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|_2^2 = (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}) = \mathbf{x}^T A^T A \mathbf{x} - 2\mathbf{b}^T A \mathbf{x} + \mathbf{b}^T \mathbf{b}$. Computing the gradient:

$$\nabla f(\mathbf{x}) = 2A^T A \mathbf{x} - 2A^T \mathbf{b} = 2A^T(A\mathbf{x} - \mathbf{b}).$$

Setting $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$:

$$A^T A \bar{\mathbf{x}} = A^T \mathbf{b}.$$

These are the **normal equations**. The Hessian is $H = 2A^T A$, which is positive semidefinite (since $\mathbf{z}^T A^T A \mathbf{z} = \|A\mathbf{z}\|_2^2 \geq 0$ for all \mathbf{z}). Therefore, f is convex, and any solution to the normal equations is a global minimum. If $A^T A$ is invertible (which occurs when A has full column rank), the unique solution is:

$$\bar{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}.$$

5.7 Differentiable Convex Functions

5.7.1 The Gradient of a Multivariate Function

Definition 5.2 (Differentiability). Let $S \subseteq \mathbb{R}^n$ be nonempty. A function $f : S \rightarrow \mathbb{R}$ is said to be **differentiable** at $\bar{\mathbf{x}} \in \text{int}(S)$ if there exists

a gradient vector $\nabla f(\bar{\mathbf{x}})$ and a function $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$ such that

$$f(\mathbf{x}) = \underbrace{f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}})}_{\text{First-order Taylor approximation}} + \|\mathbf{x} - \bar{\mathbf{x}}\| \alpha(\bar{\mathbf{x}}; \mathbf{x} - \bar{\mathbf{x}}) \quad \text{for each } \mathbf{x} \in S,$$

where $\lim_{\mathbf{x} \rightarrow \bar{\mathbf{x}}} \alpha(\bar{\mathbf{x}}; \mathbf{x} - \bar{\mathbf{x}}) = 0$.

Remark 5.3. If f is differentiable at $\bar{\mathbf{x}}$, there is a unique gradient vector given by

$$\nabla f(\bar{\mathbf{x}}) = \left(\frac{\partial f(\bar{\mathbf{x}})}{\partial x_1}, \dots, \frac{\partial f(\bar{\mathbf{x}})}{\partial x_n} \right) \in \mathbb{R}^n.$$

5.7.2 Characterization of Differentiable Convex Functions

Lemma 5.7 (Unique Subgradient for Differentiable Convex Functions). *Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be a convex function. Suppose f is differentiable at $\bar{\mathbf{x}} \in \text{int}(S)$. Then f has a unique subgradient $\nabla f(\bar{\mathbf{x}})$ at $\bar{\mathbf{x}}$, i.e., $\partial f(\bar{\mathbf{x}}) = \{\nabla f(\bar{\mathbf{x}})\}$.*

Theorem 5.8 (First-Order Characterization of Convexity). *Let S be a nonempty open convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be differentiable on S . Then f is convex if and only if for any $\bar{\mathbf{x}} \in S$:*

$$f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \quad \text{for each } \mathbf{x} \in S.$$

Remark 5.4. The affine function $f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}})$ bounds f from below. This property can be used to construct polyhedral outer-approximations of convex functions.

5.7.3 Directional Derivatives of Convex Functions

Definition 5.3 (Directional Derivative). Let $S \subseteq \mathbb{R}^n$ be a nonempty set and $f : S \rightarrow \mathbb{R}$. Let $\bar{\mathbf{x}} \in S$ and $\mathbf{d} \in \mathbb{R}^n$ be a nonzero vector such that $\bar{\mathbf{x}} + \lambda \mathbf{d} \in S$ for $\lambda > 0$ sufficiently small.

The **directional derivative of f at $\bar{\mathbf{x}}$ along the vector \mathbf{d}** is defined

as

$$f'(\bar{\mathbf{x}}; \mathbf{d}) = \lim_{\lambda \rightarrow 0^+} \frac{f(\bar{\mathbf{x}} + \lambda \mathbf{d}) - f(\bar{\mathbf{x}})}{\lambda}.$$

Lemma 5.9 (Existence of Directional Derivatives). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function. Consider any point $\bar{\mathbf{x}} \in \mathbb{R}^n$ and a nonzero direction $\mathbf{d} \in \mathbb{R}^n$. Then the directional derivative $f'(\bar{\mathbf{x}}; \mathbf{d})$ of f at $\bar{\mathbf{x}}$ in the direction \mathbf{d} exists. Moreover, if f is differentiable at $\bar{\mathbf{x}}$, we have:*

$$f'(\bar{\mathbf{x}}; \mathbf{d}) = \nabla f(\bar{\mathbf{x}})^T \mathbf{d}.$$

5.7.4 Taylor Series Approximations

Given a smooth univariate function $f : \mathbb{R} \rightarrow \mathbb{R}$, we can approximate it with a polynomial around $\bar{x} = a$ as follows:

$$f(x) \approx f(a) + f'(a)(x - a) + \frac{1}{2!} f''(a)(x - a)^2 + \frac{1}{3!} f'''(a)(x - a)^3 + \dots$$

For a smooth multivariate function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and a point $\bar{\mathbf{x}} \in \mathbb{R}^n$:

- **First-order (linear) approximation:**

$$f(\mathbf{x}) \approx f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}})$$

- **Second-order approximation:**

$$f(\mathbf{x}) \approx f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}})^T H(\bar{\mathbf{x}}) (\mathbf{x} - \bar{\mathbf{x}})$$

where the **Hessian matrix** $H(\bar{\mathbf{x}}) = \nabla^2 f(\bar{\mathbf{x}})$ is defined by $H_{ij}(\bar{\mathbf{x}}) = \frac{\partial^2 f(\bar{\mathbf{x}})}{\partial x_i \partial x_j}$ for all $i, j \in [n]$.

5.8 Twice Differentiable Convex Functions

5.8.1 Twice Differentiable Functions

Definition 5.4 (Twice Differentiability). Let S be a nonempty set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$. Then f is said to be **twice differentiable** at $\bar{\mathbf{x}} \in \text{int}(S)$ if there exists

- a gradient vector $\nabla f(\bar{\mathbf{x}})$,
- an $n \times n$ symmetric Hessian matrix $H(\bar{\mathbf{x}}) = \nabla^2 f(\bar{\mathbf{x}})$, and
- a function $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$

such that for each $\mathbf{x} \in S$:

$$f(\mathbf{x}) = f(\bar{\mathbf{x}}) + \underbrace{\nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}})^T H(\bar{\mathbf{x}}) (\mathbf{x} - \bar{\mathbf{x}})}_{\text{Second-order Taylor approximation}} + \|\mathbf{x} - \bar{\mathbf{x}}\|^2 \alpha(\bar{\mathbf{x}}; \mathbf{x} - \bar{\mathbf{x}}).$$

Example 5.6 (Computing Taylor Approximations). Consider the following functions:

1. $f(\mathbf{x}) = 4x_1 + 2x_2 - 7x_1^2 - 3x_2^2 + 5x_1x_2$ at $\bar{\mathbf{x}} = \mathbf{0}$.

The gradient is $\nabla f(\mathbf{x}) = (4 - 14x_1 + 5x_2, 2 - 6x_2 + 5x_1)^T$, so $\nabla f(\mathbf{0}) = (4, 2)^T$.

The Hessian is $H = \begin{pmatrix} -14 & 5 \\ 5 & -6 \end{pmatrix}$, which is constant.

2. $f(\mathbf{x}) = e^{2x_1+3x_2}$ at $\bar{\mathbf{x}} = (2, 1)$.

The gradient is $\nabla f(\mathbf{x}) = e^{2x_1+3x_2}(2, 3)^T$, so $\nabla f(2, 1) = e^7(2, 3)^T$.

The Hessian is $H(\mathbf{x}) = e^{2x_1+3x_2} \begin{pmatrix} 4 & 6 \\ 6 & 9 \end{pmatrix}$, so $H(2, 1) = e^7 \begin{pmatrix} 4 & 6 \\ 6 & 9 \end{pmatrix}$.

5.8.2 Positive Semidefinite Matrices

Definition 5.5 (Definiteness of Symmetric Matrices). A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is said to be:

- **Positive Semidefinite (PSD):** if $\mathbf{z}^T A \mathbf{z} \geq 0$ for all $\mathbf{z} \in \mathbb{R}^n$.
- **Positive Definite (PD):** if $\mathbf{z}^T A \mathbf{z} > 0$ for all $\mathbf{z} \in \mathbb{R}^n$ with $\mathbf{z} \neq \mathbf{0}$.
- **Negative Semidefinite (NSD):** if $\mathbf{z}^T A \mathbf{z} \leq 0$ for all $\mathbf{z} \in \mathbb{R}^n$.
- **Negative Definite (ND):** if $\mathbf{z}^T A \mathbf{z} < 0$ for all $\mathbf{z} \in \mathbb{R}^n$ with $\mathbf{z} \neq \mathbf{0}$.
- **Indefinite:** if A is neither PSD nor NSD.

Example 5.7 (Checking Definiteness). Consider the following matrices:

1. The 2×2 identity matrix I is positive definite since $\mathbf{z}^T I \mathbf{z} = \|\mathbf{z}\|_2^2 > 0$ for all $\mathbf{z} \neq \mathbf{0}$.
2. The matrix $A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$ is positive definite. This can be verified by computing its eigenvalues: $\lambda_1 = 2 - \sqrt{2}$, $\lambda_2 = 2$, $\lambda_3 = 2 + \sqrt{2}$, all of which are positive.

Lemma 5.10 (2×2 Positive Semidefiniteness). *Consider a symmetric matrix $A = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \in \mathbb{R}^{2 \times 2}$. Then A is positive semidefinite if and only if $a \geq 0$, $c \geq 0$, and $ac - b^2 \geq 0$. It is positive definite if and only if all of these inequalities are strict.*

Theorem 5.11 (Eigenvalue Characterization of Definiteness). *Consider symmetric $A \in \mathbb{R}^{n \times n}$ with eigenvalues $\lambda_1, \dots, \lambda_n \in \mathbb{R}$. Then A is:*

- **Positive semidefinite if and only if $\lambda_i \geq 0$ for each $i \in [n]$.**
- **Positive definite if and only if $\lambda_i > 0$ for each $i \in [n]$.**
- **Negative semidefinite if and only if $\lambda_i \leq 0$ for each $i \in [n]$.**

- **Negative definite** if and only if $\lambda_i < 0$ for each $i \in [n]$.
- **Indefinite** if and only if there exist $i, j \in [n]$ with $\lambda_i > 0$ and $\lambda_j < 0$.

Remark 5.5. In practice, positive definiteness of A is typically checked by computing the Cholesky factorization. If A is positive definite, there exists a unique lower triangular matrix L with positive diagonal entries such that $A = LL^T$. If the Cholesky factorization fails, A is not positive definite.

5.8.3 Characterization of Twice Differentiable Convex Functions

Theorem 5.12 (Second-Order Characterization of Convexity). *Let S be a nonempty open convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be twice differentiable on S . Then f is convex if and only if the Hessian matrix $H(\bar{\mathbf{x}}) = \nabla^2 f(\bar{\mathbf{x}})$ is positive semidefinite at each $\bar{\mathbf{x}} \in S$.*

Remark 5.6. If the function is quadratic, the Hessian matrix is independent of the point under consideration. Hence, checking its convexity reduces to checking the positive semidefiniteness of a constant matrix.

Theorem 5.13 (Strict Convexity and Positive Definiteness). *Let S be a nonempty open convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be twice differentiable on S .*

1. *If the Hessian matrix is positive definite at each point in S , then f is strictly convex.*
2. *Conversely, if f is strictly convex, the Hessian matrix is positive semidefinite at each point in S .*
3. *If f is strictly convex and quadratic, its Hessian is positive definite.*

5.8.4 Examples of Convex and Concave Functions

Example 5.8 (Convex Functions). The following are examples of convex functions:

- **Exponential function:** $\exp(x)$ on $X = \mathbb{R}$. (Strictly convex)
- **Power functions:** x^p on $X = (0, +\infty)$ for $p \geq 1$ or $p \leq 0$.
- **Negative entropy:** $x \ln(x)$ on $X = (0, +\infty)$. (Strictly convex)
- **Quadratic functions:** $\mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$ on $X = \mathbb{R}^n$ for any $A \in \mathcal{S}_+^n$, $\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$. (Strictly convex if A is positive definite)

Example 5.9 (Concave Functions). The following are examples of concave functions:

- **Power functions:** x^p on $X = (0, +\infty)$ for $p \in [0, 1]$.
- **Geometric mean:** $(\prod_{i=1}^n x_i)^{1/n}$ on $X = \mathbb{R}_{++}^n$.
- **Logarithm:** $\ln(x)$ on $X = (0, +\infty)$. (Strictly concave)
- **Quadratic functions:** $\mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$ on $X = \mathbb{R}^n$ for $-A \in \mathcal{S}_+^n$, $\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$. (Strictly concave if A is negative definite)

Chapter 6

Differentiable Convex Functions

This chapter develops the theory of differentiable convex functions, which provides powerful tools for characterizing convexity and optimality in continuous optimization. We begin with the definition of differentiability and the gradient of multivariate functions, then establish the fundamental characterization of differentiable convex functions through their first-order Taylor approximations. We also study directional derivatives and their connection to gradients. The chapter then extends to twice differentiable functions, introducing the Hessian matrix and its role in characterizing convexity. Central to this analysis is the theory of positive semidefinite matrices, including their eigenvalue characterization. The chapter concludes with necessary and sufficient conditions for optimality in convex optimization problems.

Recommended Reading

- Sections 3.1 and 3.3 of Bazaraa, Sherali, and Shetty (2006)
- Section 3.1 of Boyd and Vandenberghe

6.1 The Gradient of a Multivariate Function

We begin by formalizing the notion of differentiability for functions of several variables and introducing the gradient vector.

Definition 6.1 (Differentiable Function). Let $S \subseteq \mathbb{R}^n$ be nonempty. A function $f : S \rightarrow \mathbb{R}$ is said to be **differentiable** at $\bar{\mathbf{x}} \in \text{int}(S)$ if there exists a gradient vector $\nabla f(\bar{\mathbf{x}})$ and a function $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$ such that

$$f(\mathbf{x}) = \underbrace{f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}})}_{\text{First-order Taylor series approximation of } f \text{ near } \bar{\mathbf{x}}} + \|\mathbf{x} - \bar{\mathbf{x}}\| \alpha(\bar{\mathbf{x}}; \mathbf{x} - \bar{\mathbf{x}}) \quad \text{for each } \mathbf{x} \in S,$$

(6.1)

where $\lim_{\mathbf{x} \rightarrow \bar{\mathbf{x}}} \alpha(\bar{\mathbf{x}}; \mathbf{x} - \bar{\mathbf{x}}) = 0$.

Remark 6.1 (Uniqueness of the Gradient). If f is differentiable at $\bar{\mathbf{x}}$, there is a *unique* **gradient vector** given by

$$\nabla f(\bar{\mathbf{x}}) = \left(\frac{\partial f(\bar{\mathbf{x}})}{\partial x_1}, \dots, \frac{\partial f(\bar{\mathbf{x}})}{\partial x_n} \right) \in \mathbb{R}^n. \quad (6.2)$$

The gradient vector collects all the partial derivatives of f evaluated at $\bar{\mathbf{x}}$.

The gradient has a natural geometric interpretation: it points in the direction of steepest ascent of f at the point $\bar{\mathbf{x}}$, and its magnitude indicates the rate of increase in that direction.

6.2 Characterization of Differentiable Convex Functions

One of the most important results in convex analysis establishes the connection between convexity and the gradient. We first show that for differentiable convex functions, the gradient is the unique subgradient.

Lemma 6.1 (Gradient is the Unique Subgradient). *Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be a convex function. Suppose f is differentiable at $\bar{\mathbf{x}} \in \text{int}(S)$. Then f has a unique subgradient $\nabla f(\bar{\mathbf{x}})$ at $\bar{\mathbf{x}}$, i.e., $\partial f(\bar{\mathbf{x}}) = \{\nabla f(\bar{\mathbf{x}})\}$.*

The following theorem provides a fundamental characterization of differentiable convex functions in terms of their first-order approximation.

Theorem 6.2 (First-Order Characterization of Convexity). *Let S be a nonempty open convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be differentiable on S . Then f is convex if and only if for any $\bar{\mathbf{x}} \in S$:*

$$f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \quad \text{for each } \mathbf{x} \in S. \quad (6.3)$$

Remark 6.2 (Geometric Interpretation). The inequality (6.3) states that for a differentiable convex function, the first-order Taylor approximation (the tangent hyperplane at any point) provides a global under-estimator of the function. Geometrically, the graph of f lies above all of its tangent hyperplanes.

Remark 6.3 (Applications to Optimization). The affine function $f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}})$ bounds f from below. This property is fundamental for constructing polyhedral outer-approximations of convex functions, which forms the basis for cutting-plane methods and outer-approximation algorithms in convex optimization.

6.3 Directional Derivatives of Convex Functions

Directional derivatives generalize the notion of the derivative to arbitrary directions, not just along coordinate axes.

Definition 6.2 (Directional Derivative). Let $S \subseteq \mathbb{R}^n$ be a nonempty set and $f : S \rightarrow \mathbb{R}$. Let $\bar{\mathbf{x}} \in S$ and $\mathbf{d} \in \mathbb{R}^n$ be a nonzero vector such that $\bar{\mathbf{x}} + \lambda\mathbf{d} \in S$ for $\lambda > 0$ sufficiently small.

The **directional derivative of f at $\bar{\mathbf{x}}$ along the vector \mathbf{d}** is defined as

$$f'(\bar{\mathbf{x}}; \mathbf{d}) = \lim_{\lambda \rightarrow 0^+} \frac{f(\bar{\mathbf{x}} + \lambda\mathbf{d}) - f(\bar{\mathbf{x}})}{\lambda}. \quad (6.4)$$

Lemma 6.3 (Existence of Directional Derivatives). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function. Consider any point $\bar{\mathbf{x}} \in \mathbb{R}^n$ and a nonzero direction $\mathbf{d} \in \mathbb{R}^n$. Then the directional derivative $f'(\bar{\mathbf{x}}; \mathbf{d})$ of f at $\bar{\mathbf{x}}$ in the direction \mathbf{d} exists.*

Moreover, if f is differentiable at $\bar{\mathbf{x}}$, we have:

$$f'(\bar{\mathbf{x}}; \mathbf{d}) = \nabla f(\bar{\mathbf{x}})^T \mathbf{d}. \quad (6.5)$$

Remark 6.4. The formula (6.5) shows that for differentiable functions, the directional derivative in any direction \mathbf{d} can be computed as the inner product of the gradient with the direction vector. This explains why the gradient points in the direction of steepest ascent: the directional derivative is maximized when \mathbf{d} is aligned with $\nabla f(\bar{\mathbf{x}})$.

6.4 Taylor Series Approximations

Taylor series provide polynomial approximations to smooth functions near a given point. We review both univariate and multivariate cases.

6.4.1 Univariate Taylor Series

Given a smooth univariate function $f : \mathbb{R} \rightarrow \mathbb{R}$, we can approximate it with a polynomial around $\bar{x} = a$ as follows:

$$f(x) \approx f(a) + f'(a)(x - a) + \frac{1}{2!}f''(a)(x - a)^2 + \frac{1}{3!}f'''(a)(x - a)^3 + \dots \quad (6.6)$$

6.4.2 Multivariate Taylor Series

Given a smooth multivariate function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and a point $\bar{\mathbf{x}} \in \mathbb{R}^n$, we have the following approximations:

- **Linear/First-order approximation:**

$$f(\mathbf{x}) \approx f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) \quad (6.7)$$

- **Second-order approximation:**

$$f(\mathbf{x}) \approx f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T H(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}}) \quad (6.8)$$

where the **Hessian matrix** $H(\bar{\mathbf{x}}) = \nabla^2 f(\bar{\mathbf{x}})$ is defined by

$$H_{ij}(\bar{\mathbf{x}}) = \frac{\partial^2 f(\bar{\mathbf{x}})}{\partial x_i \partial x_j}, \quad \forall i, j \in [n]. \quad (6.9)$$

6.5 Twice Differentiable Functions

We now formalize the notion of twice differentiability and introduce the Hessian matrix.

Definition 6.3 (Twice Differentiable Function). Let S be a nonempty set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$. Then f is said to be **twice differentiable** at $\bar{\mathbf{x}} \in \text{int}(S)$ if there exist:

- a gradient vector $\nabla f(\bar{\mathbf{x}})$,
- an $n \times n$ symmetric Hessian matrix $H(\bar{\mathbf{x}}) = \nabla^2 f(\bar{\mathbf{x}})$, and
- a function $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$

such that for each $\mathbf{x} \in S$:

$$f(\mathbf{x}) = f(\bar{\mathbf{x}}) + \underbrace{\nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}})^T H(\bar{\mathbf{x}}) (\mathbf{x} - \bar{\mathbf{x}})}_{\text{Second-order Taylor series approximation of } f \text{ near } \bar{\mathbf{x}}} + \|\mathbf{x} - \bar{\mathbf{x}}\|^2 \alpha(\bar{\mathbf{x}}; \mathbf{x} - \bar{\mathbf{x}}), \quad (6.10)$$

where $\lim_{\mathbf{x} \rightarrow \bar{\mathbf{x}}} \alpha(\bar{\mathbf{x}}; \mathbf{x} - \bar{\mathbf{x}}) = 0$.

Example 6.1 (Computing Gradient and Hessian). Consider the following functions:

1. Let $f(\mathbf{x}) = 4x_1 + 2x_2 - 7x_1^2 - 3x_2^2 + 5x_1x_2$ at $\bar{\mathbf{x}} = \mathbf{0}$.

The gradient is:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} 4 - 14x_1 + 5x_2 \\ 2 - 6x_2 + 5x_1 \end{pmatrix}, \quad \nabla f(\mathbf{0}) = \begin{pmatrix} 4 \\ 2 \end{pmatrix}.$$

The Hessian is:

$$H(\mathbf{x}) = \begin{pmatrix} -14 & 5 \\ 5 & -6 \end{pmatrix},$$

which is constant (independent of \mathbf{x}) since f is quadratic.

2. Let $f(\mathbf{x}) = e^{2x_1+3x_2}$ at $\bar{\mathbf{x}} = (2, 1)$.

The gradient is:

$$\nabla f(\mathbf{x}) = e^{2x_1+3x_2} \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \quad \nabla f(2, 1) = e^7 \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

The Hessian is:

$$H(\mathbf{x}) = e^{2x_1+3x_2} \begin{pmatrix} 4 & 6 \\ 6 & 9 \end{pmatrix}, \quad H(2, 1) = e^7 \begin{pmatrix} 4 & 6 \\ 6 & 9 \end{pmatrix}.$$

6.6 Positive Semidefinite Matrices

The definiteness of the Hessian matrix plays a crucial role in characterizing the convexity of twice differentiable functions. We now introduce the relevant definitions.

Definition 6.4 (Matrix Definiteness). A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is said to be:

- **Positive Semidefinite (PSD):** if $\mathbf{z}^T \mathbf{A} \mathbf{z} \geq 0$ for all $\mathbf{z} \in \mathbb{R}^n$. We write $\mathbf{A} \succeq 0$.
- **Positive Definite (PD):** if $\mathbf{z}^T \mathbf{A} \mathbf{z} > 0$ for all $\mathbf{z} \in \mathbb{R}^n$ with $\mathbf{z} \neq \mathbf{0}$. We write $\mathbf{A} \succ 0$.
- **Negative Semidefinite (NSD):** if $\mathbf{z}^T \mathbf{A} \mathbf{z} \leq 0$ for all $\mathbf{z} \in \mathbb{R}^n$. We write $\mathbf{A} \preceq 0$.
- **Negative Definite (ND):** if $\mathbf{z}^T \mathbf{A} \mathbf{z} < 0$ for all $\mathbf{z} \in \mathbb{R}^n$ with $\mathbf{z} \neq \mathbf{0}$. We write $\mathbf{A} \prec 0$.
- **Indefinite:** if \mathbf{A} is neither positive semidefinite nor negative semidefinite.

Example 6.2 (Matrix Definiteness). 1. The 2×2 identity matrix

$\mathbf{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ is positive definite since for any $\mathbf{z} = (z_1, z_2)^T \neq \mathbf{0}$:

$$\mathbf{z}^T \mathbf{I}_2 \mathbf{z} = z_1^2 + z_2^2 > 0.$$

2. Consider the matrix

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}.$$

This is a tridiagonal matrix. To determine its definiteness, we can compute its eigenvalues or check the signs of its leading principal minors.

6.6.1 Checking Positive Semidefiniteness

For 2×2 matrices, there is a simple algebraic criterion.

Lemma 6.4 (2×2 Matrix Criterion). *Consider a symmetric matrix $\mathbf{A} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \in \mathbb{R}^{2 \times 2}$. Then:*

- \mathbf{A} is positive semidefinite if and only if $a \geq 0$, $c \geq 0$, and $ac - b^2 \geq 0$.
- \mathbf{A} is positive definite if and only if all of these inequalities are strict.

For general $n \times n$ matrices, the eigenvalue characterization is fundamental.

Theorem 6.5 (Eigenvalue Characterization of Definiteness). *Consider a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ with eigenvalues $\lambda_1, \dots, \lambda_n \in \mathbb{R}$. Then:*

- \mathbf{A} is positive semidefinite if and only if $\lambda_i \geq 0$ for each $i \in [n]$.
- \mathbf{A} is positive definite if and only if $\lambda_i > 0$ for each $i \in [n]$.
- \mathbf{A} is negative semidefinite if and only if $\lambda_i \leq 0$ for each $i \in [n]$.
- \mathbf{A} is negative definite if and only if $\lambda_i < 0$ for each $i \in [n]$.
- \mathbf{A} is indefinite if and only if there exist $i, j \in [n]$ with $\lambda_i > 0$ and $\lambda_j < 0$.

Remark 6.5 (Computational Practice). In practice, positive definiteness of a matrix \mathbf{A} is typically checked by computing its **Cholesky factorization**. The matrix \mathbf{A} is positive definite if and only if the Cholesky factorization $\mathbf{A} = \mathbf{L}\mathbf{L}^T$ exists, where \mathbf{L} is a lower triangular matrix with positive diagonal entries. This approach is numerically

stable and has computational complexity $O(n^3)$.

6.7 Characterization of Twice Differentiable Convex Functions

We now state the main result connecting the Hessian matrix to convexity.

Theorem 6.6 (Second-Order Characterization of Convexity). *Let S be a nonempty open convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be twice differentiable on S . Then f is convex if and only if the Hessian matrix $H(\bar{\mathbf{x}}) = \nabla^2 f(\bar{\mathbf{x}})$ is positive semidefinite at each $\bar{\mathbf{x}} \in S$.*

Remark 6.6 (Quadratic Functions). If the function f is quadratic, i.e., $f(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} + d$ for some symmetric matrix \mathbf{Q} , then the Hessian matrix $H(\mathbf{x}) = 2\mathbf{Q}$ is constant (independent of the point under consideration). Hence, checking the convexity of a quadratic function reduces to checking the positive semidefiniteness of a constant matrix.

The following theorem addresses the relationship between the Hessian and strict convexity.

Theorem 6.7 (Strict Convexity and the Hessian). *Let S be a nonempty open convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be twice differentiable on S .*

1. *If the Hessian matrix is positive definite at each point in S , then f is strictly convex.*
2. *Conversely, if f is strictly convex, the Hessian matrix is positive semidefinite at each point in S .*
3. *If f is strictly convex and quadratic, its Hessian is positive definite.*

Remark 6.7. Note the asymmetry in the theorem: positive definiteness of the Hessian implies strict convexity, but strict convexity only implies positive semidefiniteness in general. The classic counterexample is $f(x) = x^4$ on \mathbb{R} , which is strictly convex but has $f''(0) = 0$.

6.8 Examples of Convex and Concave Functions

We collect some important examples of convex and concave functions, many of which can be verified using the second-order characterization.

6.8.1 Examples of Convex Functions

- **Exponential function:** $\exp(x)$ on $X = \mathbb{R}$. (Strictly convex since $f''(x) = e^x > 0$)
- **Power functions:** x^p on $X = (0, +\infty)$ for $p \geq 1$ or $p \leq 0$.
- **Negative entropy:** $x \ln(x)$ on $X = (0, +\infty)$. (Strictly convex since $f''(x) = 1/x > 0$)
- **Quadratic functions:** $\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$ on $X = \mathbb{R}^n$ for any $\mathbf{A} \in \mathcal{S}_+^n$, $\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$. (Strictly convex if \mathbf{A} is positive definite)

6.8.2 Examples of Concave Functions

- **Power functions:** x^p on $X = (0, +\infty)$ for $p \in [0, 1]$.
- **Geometric mean:** $(\prod_{i=1}^n x_i)^{1/n}$ on $X = \mathbb{R}_{++}^n$.
- **Logarithm:** $\ln(x)$ on $X = (0, +\infty)$. (Strictly concave since $f''(x) = -1/x^2 < 0$)
- **Quadratic functions:** $\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$ on $X = \mathbb{R}^n$ for $-\mathbf{A} \in \mathcal{S}_+^n$, $\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$. (Strictly concave if \mathbf{A} is negative definite)

6.9 Recap: Minimizing a Convex Function

We now review the key results on convex optimization that utilize the concepts developed in this chapter.

Definition 6.5 (Convex Optimization Problem). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function and S be a nonempty convex set in \mathbb{R}^n . A **convex optimization problem** is defined by

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in S. \end{aligned} \tag{6.11}$$

Theorem 6.8 (Local Optimality Implies Global Optimality). *Suppose that $\bar{\mathbf{x}} \in S$ is a local optimal solution to Problem (6.11).*

1. *Then $\bar{\mathbf{x}}$ is a global optimal solution.*
2. *If either $\bar{\mathbf{x}}$ is a strict local minimum or f is strictly convex, then $\bar{\mathbf{x}}$ is the unique global optimal solution (it is also a strong local minimum).*

6.9.1 Necessary and Sufficient Conditions for a Global Minimum

Theorem 6.9 (Subgradient Optimality Condition). *Point $\bar{\mathbf{x}} \in S$ is an optimal solution to Problem (6.11) if and only if f has a subgradient ξ at $\bar{\mathbf{x}}$ such that*

$$\xi^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0 \quad \text{for all } \mathbf{x} \in S. \quad (6.12)$$

Corollary 6.10 (Zero Subgradient Condition). *Under the assumptions of Theorem 6.9, if S is an open set (for example, $S = \mathbb{R}^n$), then $\bar{\mathbf{x}}$ is an optimal solution to the problem if and only if there exists a zero subgradient of f at $\bar{\mathbf{x}}$.*

Corollary 6.11 (Gradient Optimality Condition). *Assume that f is differentiable. Then $\bar{\mathbf{x}}$ is an optimal solution if and only if*

$$\nabla f(\bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \geq 0 \quad \text{for all } \mathbf{x} \in S. \quad (6.13)$$

Furthermore, if S is open, $\bar{\mathbf{x}}$ is an optimal solution if and only if $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$.

6.9.2 Examples

Example 6.3 (Unconstrained Optimization). Consider the following optimization problems:

1. Problem 1:

$$\begin{aligned} \min \quad & x_1 + x_1 x_2 + x_2^2 \\ \text{s.t.} \quad & (x_1, x_2) \in \mathbb{R}^2. \end{aligned}$$

Let $f(x_1, x_2) = x_1 + x_1 x_2 + x_2^2$. The gradient is:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} 1 + x_2 \\ x_1 + 2x_2 \end{pmatrix}.$$

Setting $\nabla f(\mathbf{x}) = \mathbf{0}$ gives $x_2 = -1$ and $x_1 = 2$.

The Hessian is:

$$H = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}.$$

Since $\det(H) = -1 < 0$, the Hessian is indefinite. Thus f is not convex, and the stationary point $(2, -1)$ is a saddle point, not a minimum. This problem is unbounded below.

2. Problem 2:

$$\begin{aligned} \min \quad & x_1 + x_1^2 + x_2^2 \\ \text{s.t.} \quad & (x_1, x_2) \in \mathbb{R}^2. \end{aligned}$$

Let $f(x_1, x_2) = x_1 + x_1^2 + x_2^2$. The gradient is:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} 1 + 2x_1 \\ 2x_2 \end{pmatrix}.$$

Setting $\nabla f(\mathbf{x}) = \mathbf{0}$ gives $x_1 = -1/2$ and $x_2 = 0$.

The Hessian is:

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = 2\mathbf{I}_2,$$

which is positive definite. Thus f is strictly convex, and $\bar{\mathbf{x}} = (-1/2, 0)$ is the unique global minimizer with optimal value $f(\bar{\mathbf{x}}) = -1/4$.

Chapter 7

Generalizations of Convexity

This chapter explores various generalizations of convexity that extend the powerful properties of convex functions to broader classes. We begin with convex maximization and its surprising connection to extreme point solutions, then systematically develop the theory of quasiconvex, strictly quasiconvex, strongly quasiconvex, strongly convex, and pseudoconvex functions. These generalizations preserve certain desirable properties of convex functions—such as local-to-global optimality or uniqueness of solutions—while relaxing the strict requirements of convexity.

Recommended Reading

- Sections 3.3, 3.4, and 3.5 of Bazaraa, Sherali, and Shetty (2006)
- Section 3.4 of Boyd and Vandenberghe

7.1 Convex Maximization

While convex minimization problems enjoy the property that every local minimum is a global minimum, the situation is quite different for convex maximization. In this section, we study the problem of maximizing a convex function over a convex set and establish conditions under which optimal solutions can be characterized.

7.1.1 Necessary Conditions for Local Maxima

Theorem 7.1 (Necessary Condition for Local Maximum). *Let S be a nonempty convex set in \mathbb{R}^n and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function. If $\mathbf{x}^* \in S$ is a local maximum to the problem*

$$\max_{\mathbf{x} \in S} f(\mathbf{x}),$$

then

$$\boldsymbol{\xi}^T(\mathbf{x} - \mathbf{x}^*) \leq 0 \quad \text{for all } \mathbf{x} \in S,$$

where $\boldsymbol{\xi}$ is any subgradient of f at \mathbf{x}^ .*

Corollary 7.2. *Suppose that f is also differentiable. If $\mathbf{x}^* \in S$ is a local maximum, then $\nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) \leq 0$ for all $\mathbf{x} \in S$.*

Remark 7.1. These conditions are necessary but not sufficient for optimality. For example, consider $f(x) = x^2$ and $S := \{x \in \mathbb{R} : -1 \leq x \leq 2\}$. The point $\bar{x} = 0$ satisfies the necessary condition (since $\nabla f(0) = 0$), but it is clearly not a local maximum—in fact, it is the global minimum on S .

7.1.2 Extreme Point Solutions

One of the most remarkable properties of convex maximization over polyhedral sets is that an optimal solution always exists at an extreme point. This generalizes the fundamental theorem of linear programming.

Theorem 7.3 (Convex Maximization Has an Extreme Point Solution). *Let S be a nonempty compact polyhedral set in \mathbb{R}^n and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be convex. The problem $\max_{\mathbf{x} \in S} f(\mathbf{x})$ has an optimal solution that is an extreme point of S .*

This result is a generalization of the fundamental theorem of linear programming, which states that if a linear program has an optimal solution, then it has an optimal solution at an extreme point. The theorem can also be generalized to non-polyhedral convex sets S .

Example 7.1. Let $f(\mathbf{x}) := (x_1 - \frac{3}{2})^2 + (x_2 - 5)^2$ and

$$S := \{\mathbf{x} \in \mathbb{R}_+^2 : -x_1 + x_2 \leq 2, 2x_1 + 3x_2 \leq 11\}.$$

1. What is $\min\{f(\mathbf{x}) : \mathbf{x} \in S\}$?

Since f is a convex quadratic function, the minimum over the convex set S can occur either at an interior point (if the unconstrained minimizer lies in S) or on the boundary. The unconstrained minimizer is $\mathbf{x}^* = (3/2, 5)^T$, but this point does not lie in S . Therefore, the minimum is attained on the boundary of S .

2. What is $\max\{f(\mathbf{x}) : \mathbf{x} \in S\}$?

By the theorem above, the maximum must occur at an extreme point of S . The extreme points of S are $(0, 0)$, $(0, 2)$, $(1, 3)$, and $(5.5, 0)$. Evaluating f at each extreme point:

- $f(0, 0) = (0 - 3/2)^2 + (0 - 5)^2 = 2.25 + 25 = 27.25$
- $f(0, 2) = (0 - 3/2)^2 + (2 - 5)^2 = 2.25 + 9 = 11.25$
- $f(1, 3) = (1 - 3/2)^2 + (3 - 5)^2 = 0.25 + 4 = 4.25$
- $f(5.5, 0) = (5.5 - 3/2)^2 + (0 - 5)^2 = 16 + 25 = 41$

Thus, the maximum value is 41, attained at the extreme point $(5.5, 0)$.

See Figure 7.1 for an illustration.

7.2 Quasiconvex and Quasiconcave Functions

Quasiconvex functions represent one of the most important generalizations of convexity. While they do not preserve all the nice properties of convex functions, they maintain a crucial structural property: convexity of level sets.

Definition 7.1 (Quasiconvex Function). Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$. The function f is said to be **quasiconvex** if

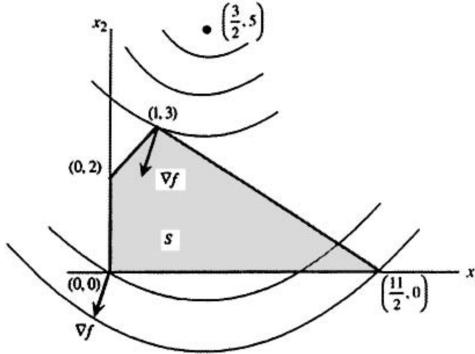


Figure 7.1: Illustration of convex maximization over a polyhedral set S . The contours of the convex quadratic function $f(\mathbf{x}) = (x_1 - 3/2)^2 + (x_2 - 5)^2$ are shown, with the unconstrained minimizer at $(3/2, 5)$ lying outside S . The maximum is attained at the extreme point $(11/2, 0)$.

for each $\mathbf{x}_1, \mathbf{x}_2 \in S$ and $\lambda \in (0, 1)$:

$$f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \leq \max\{f(\mathbf{x}_1), f(\mathbf{x}_2)\}.$$

The function f is said to be **quasiconcave** if $-f$ is quasiconvex.

The defining property of quasiconvexity states that the function value at any convex combination of two points does not exceed the larger of the two function values. This is a weaker condition than convexity, which requires the function value to be bounded by the convex combination of the function values.

7.2.1 Lower Level Sets of Quasiconvex Functions

The most elegant characterization of quasiconvexity is in terms of the convexity of lower level sets.

Definition 7.2 (Lower Level Set). Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$. For each $\alpha \in \mathbb{R}$, the **lower level set** (or **sublevel set**) of f is defined as

$$S_\alpha = \{\mathbf{x} \in S : f(\mathbf{x}) \leq \alpha\}.$$

Theorem 7.4 (Characterization via Lower Level Sets). *Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$. The function f is quasiconvex if and only if the lower level set $S_\alpha = \{\mathbf{x} \in S : f(\mathbf{x}) \leq \alpha\}$ is convex for each $\alpha \in \mathbb{R}$.*

Proof. (\Rightarrow) Suppose f is quasiconvex. Let $\alpha \in \mathbb{R}$ and take any $\mathbf{x}_1, \mathbf{x}_2 \in S_\alpha$. Then $f(\mathbf{x}_1) \leq \alpha$ and $f(\mathbf{x}_2) \leq \alpha$. For any $\lambda \in (0, 1)$, by quasiconvexity:

$$f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \leq \max\{f(\mathbf{x}_1), f(\mathbf{x}_2)\} \leq \alpha.$$

Hence $\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in S_\alpha$, so S_α is convex.

(\Leftarrow) Suppose all lower level sets S_α are convex. Let $\mathbf{x}_1, \mathbf{x}_2 \in S$ and set $\alpha = \max\{f(\mathbf{x}_1), f(\mathbf{x}_2)\}$. Then $\mathbf{x}_1, \mathbf{x}_2 \in S_\alpha$. Since S_α is convex, for any $\lambda \in (0, 1)$, we have $\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in S_\alpha$, which means

$$f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \leq \alpha = \max\{f(\mathbf{x}_1), f(\mathbf{x}_2)\}.$$

Therefore, f is quasiconvex. □

Example 7.2 (Quasiconvex Functions). (a) $f(x) = x^3$ is quasiconvex on \mathbb{R} . The lower level set $\{x : x^3 \leq \alpha\} = (-\infty, \alpha^{1/3}]$ is convex (a half-line) for each α .

(b) $f(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$ (the standard normal density) is quasiconcave. Its upper level sets are convex (intervals centered at zero).

(c) Every convex function is also quasiconvex. Indeed, if f is convex, then for any $\mathbf{x}_1, \mathbf{x}_2 \in S$ and $\lambda \in (0, 1)$:

$$f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda)f(\mathbf{x}_2) \leq \max\{f(\mathbf{x}_1), f(\mathbf{x}_2)\}.$$

Remark 7.2. The converse of part (c) is false: not every quasiconvex function is convex. For instance, $f(x) = x^3$ is quasiconvex but not convex on \mathbb{R} .

7.2.2 Quasiconvex Maximization Has an Extreme Point Solution

The extreme point property established for convex maximization extends to quasiconvex functions.

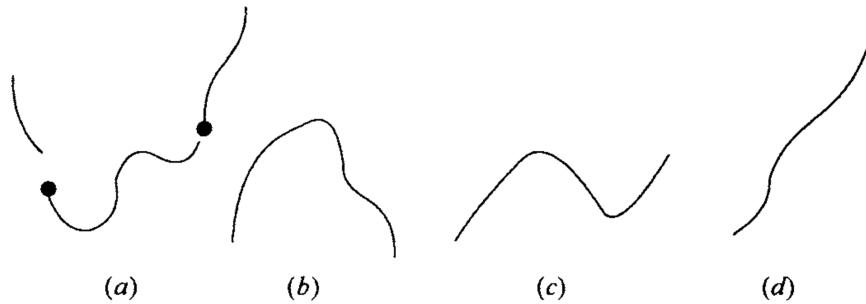


Figure 7.2: Examples illustrating quasiconvexity and local vs. global minima. (a) A quasiconvex function with a unique local minimum that is also global. (b) A strictly quasiconvex function where the local minimum (marked) is the global minimum. (c) A function that is not quasiconvex—it has a local minimum that is not global. (d) A quasiconvex function (monotonic).

Theorem 7.5 (Quasiconvex Maximization Has an Extreme Point Solution). *Let S be a nonempty compact polyhedral set in \mathbb{R}^n and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be quasiconvex and continuous on S . The problem $\max_{\mathbf{x} \in S} f(\mathbf{x})$ has an optimal solution that is an extreme point of S .*

7.3 Strictly Quasiconvex Functions

A local optimal solution to a quasiconvex minimization problem is not necessarily a global optimal solution. Moreover, a sum of quasiconvex functions is not necessarily quasiconvex. For a discussion of operations preserving quasiconvexity, see Section 3.4.4 of Boyd and Vandenberghe.

Strictly quasiconvex functions provide a remedy for the local-to-global optimality issue.

Definition 7.3 (Strictly Quasiconvex Function). Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$. The function f is said to be **strictly quasiconvex** if for each $\mathbf{x}_1, \mathbf{x}_2 \in S$ with $f(\mathbf{x}_1) \neq f(\mathbf{x}_2)$ and each $\lambda \in (0, 1)$:

$$f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) < \max\{f(\mathbf{x}_1), f(\mathbf{x}_2)\}.$$

The function f is said to be **strictly quasiconcave** if $-f$ is strictly quasiconvex.

Remark 7.3. A strictly quasiconvex function is not necessarily quasiconvex! This may seem counterintuitive, but the strict inequality in the definition only applies when $f(\mathbf{x}_1) \neq f(\mathbf{x}_2)$. When $f(\mathbf{x}_1) = f(\mathbf{x}_2)$, the definition places no restriction on the function value at the convex combination.

The following result shows that additional regularity restores the expected relationship.

Proposition 7.6. *If f is strictly quasiconvex and lower semicontinuous, then f is quasiconvex.*

The key property of strictly quasiconvex functions is the following local-to-global optimality result.

Theorem 7.7 (Local Implies Global for Strictly Quasiconvex Functions). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be strictly quasiconvex and $S \subseteq \mathbb{R}^n$ be a nonempty convex set. If \mathbf{x}^* is a local optimal solution to the problem $\min_{\mathbf{x} \in S} f(\mathbf{x})$, then \mathbf{x}^* is also a global optimal solution.*

Proof. Suppose \mathbf{x}^* is a local minimum but not a global minimum. Then there exists $\bar{\mathbf{x}} \in S$ with $f(\bar{\mathbf{x}}) < f(\mathbf{x}^*)$. Consider any $\lambda \in (0, 1)$ and let $\mathbf{x}_\lambda = \lambda \bar{\mathbf{x}} + (1 - \lambda)\mathbf{x}^*$. Since $f(\bar{\mathbf{x}}) \neq f(\mathbf{x}^*)$, by strict quasiconvexity:

$$f(\mathbf{x}_\lambda) < \max\{f(\bar{\mathbf{x}}), f(\mathbf{x}^*)\} = f(\mathbf{x}^*).$$

As $\lambda \rightarrow 0$, we have $\mathbf{x}_\lambda \rightarrow \mathbf{x}^*$. This means there are points arbitrarily close to \mathbf{x}^* with strictly smaller function values, contradicting the assumption that \mathbf{x}^* is a local minimum. \square

Remark 7.4. Unlike the case of strict convexity, minimization problems with a strictly quasiconvex objective need not have unique solutions. The set of optimal solutions can contain multiple points, though the theorem guarantees they are all global optima.

7.4 Strongly Quasiconvex Functions

Strongly quasiconvex functions strengthen the definition of strict quasiconvexity by requiring the strict inequality to hold whenever $\mathbf{x}_1 \neq \mathbf{x}_2$, regardless

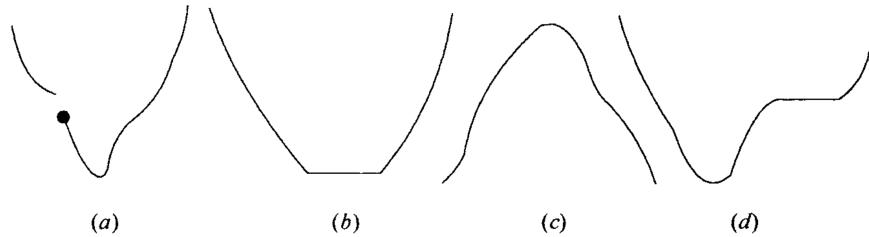


Figure 7.3: Comparison of quasiconvexity variants. (a) A strictly quasiconvex function with a unique global minimum. (b) A quasiconvex function with a flat region at the minimum—not strictly quasiconvex but still quasiconvex. (c) A function that is not quasiconvex due to multiple local minima. (d) A strictly quasiconvex function that is also strongly quasiconvex.

of whether the function values are equal.

Definition 7.4 (Strongly Quasiconvex Function). Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$. The function f is said to be **strongly quasiconvex** if for each $\mathbf{x}_1, \mathbf{x}_2 \in S$ with $\mathbf{x}_1 \neq \mathbf{x}_2$ and each $\lambda \in (0, 1)$:

$$f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) < \max\{f(\mathbf{x}_1), f(\mathbf{x}_2)\}.$$

The function f is said to be **strongly quasiconcave** if $-f$ is strongly quasiconvex.

The relationships between these function classes can be summarized as follows:

Proposition 7.8 (Hierarchy of Quasiconvexity). *The following implications hold:*

1. $\text{Strictly Convex} \Rightarrow \text{Strongly Quasiconvex}$.
2. $\text{Strongly Quasiconvex} \Rightarrow \text{Strictly Quasiconvex}$.
3. $\text{Strongly Quasiconvex} \Rightarrow \text{Quasiconvex}$.

The third implication follows because the strong quasiconvexity condition directly implies the quasiconvexity condition (the strict inequality implies the non-strict one).

The key advantage of strong quasiconvexity over strict quasiconvexity is uniqueness of optimal solutions.

Theorem 7.9 (Unique Global Minimum for Strongly Quasiconvex Functions). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be strongly quasiconvex and $S \subseteq \mathbb{R}^n$ be a nonempty convex set. If \mathbf{x}^* is a local optimal solution to the problem $\min_{\mathbf{x} \in S} f(\mathbf{x})$, then \mathbf{x}^* is the unique global optimal solution.*

Proof. From the local-to-global result for strictly quasiconvex functions (which applies here since strongly quasiconvex implies strictly quasiconvex), \mathbf{x}^* is a global minimum. Suppose there exists another global minimum $\bar{\mathbf{x}} \neq \mathbf{x}^*$ with $f(\bar{\mathbf{x}}) = f(\mathbf{x}^*)$. Consider $\mathbf{x}_\lambda = \lambda\bar{\mathbf{x}} + (1 - \lambda)\mathbf{x}^*$ for any $\lambda \in (0, 1)$. Since $\bar{\mathbf{x}} \neq \mathbf{x}^*$, by strong quasiconvexity:

$$f(\mathbf{x}_\lambda) < \max\{f(\bar{\mathbf{x}}), f(\mathbf{x}^*)\} = f(\mathbf{x}^*),$$

which contradicts the global optimality of \mathbf{x}^* . Hence, \mathbf{x}^* is the unique global minimum. \square

7.5 Strongly Convex Functions

Strong convexity is a quantitative strengthening of convexity that has important implications for the analysis of optimization algorithms. A strongly convex function is “more convex” than a quadratic, in a precise sense.

Definition 7.5 (μ -Strongly Convex Function). Let S be a nonempty convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$. The function f is said to be μ -strongly convex with $\mu > 0$ if the function $g : S \rightarrow \mathbb{R}$, defined by

$$g(\mathbf{x}) := f(\mathbf{x}) - \mu\|\mathbf{x}\|_2^2,$$

is convex.

The function f is said to be μ -strongly concave if $-f$ is μ -strongly convex.

Theorem 7.10 (Characterization of Strong Convexity via Hessian). *Let S be a nonempty open convex set in \mathbb{R}^n and $f : S \rightarrow \mathbb{R}$ be twice differentiable on S . Then f is μ -strongly convex on S if and only if the matrix $\nabla^2 f(\mathbf{x}) - 2\mu I_n$ is positive semidefinite at each point $\mathbf{x} \in S$.*

Equivalently, f is μ -strongly convex if and only if the minimum eigenvalue of $\nabla^2 f(\mathbf{x})$ is at least 2μ for all $\mathbf{x} \in S$.

- Example 7.3** (Strongly Convex Functions). (a) The quadratic function $f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$ is μ -strongly convex whenever A is positive definite and $0 < \mu \leq \lambda_{\min}(A)$, where $\lambda_{\min}(A)$ denotes the minimum eigenvalue of A .
- (b) The function $f(x) = \exp(x)$ is μ -strongly convex on $S = [a, b]$ for any $\mu \leq \frac{1}{2} \exp(a)$. Indeed, $f''(x) = \exp(x) \geq \exp(a) \geq 2\mu$ for all $x \in [a, b]$.
- (c) Consider $f(x) = x \ln(x)$ on $(0, +\infty)$. We have $f''(x) = 1/x$, which approaches 0 as $x \rightarrow +\infty$. Therefore, f is not μ -strongly convex on $(0, +\infty)$ for any $\mu > 0$. However, f is μ -strongly convex on any bounded interval $[a, b] \subset (0, +\infty)$ for $\mu \leq 1/(2b)$.

7.6 Other Generalizations of Convexity

There are several other generalizations of convexity that arise in various applications and theoretical developments.

7.6.1 Pseudoconvex Functions

Definition 7.6 (Pseudoconvex Function). Let $S \subseteq \mathbb{R}^n$ be a nonempty convex set and $f : S \rightarrow \mathbb{R}$ be differentiable on S . The function f is said to be **pseudoconvex** if for all $\mathbf{x}_1, \mathbf{x}_2 \in S$:

$$\nabla f(\mathbf{x}_1)^T (\mathbf{x}_2 - \mathbf{x}_1) \geq 0 \implies f(\mathbf{x}_2) \geq f(\mathbf{x}_1).$$

The function f is said to be **pseudconcave** if $-f$ is pseudoconvex.

The key properties of pseudoconvex functions include:

- All convex differentiable functions are pseudoconvex.
- If f is pseudoconvex and $\nabla f(\mathbf{x}^*) = \mathbf{0}$, then \mathbf{x}^* is a global minimum.
- Every pseudoconvex function is strictly quasiconvex and quasiconcave.

For a detailed treatment of pseudoconvex functions, see Section 3.5 of Bazaraa et al.

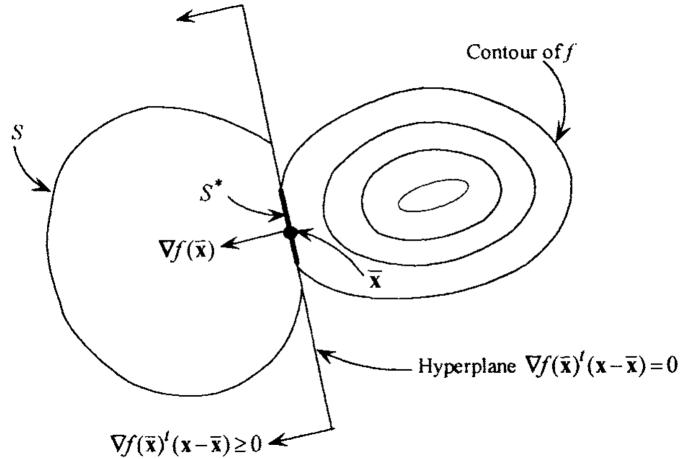


Figure 7.4: Geometric interpretation of pseudoconvexity and first-order conditions. The figure shows a pseudoconvex function f with contours (level sets) around a minimum at $\bar{\mathbf{x}}$. The gradient $\nabla f(\bar{\mathbf{x}})$ is perpendicular to the level set at $\bar{\mathbf{x}}$. The hyperplane $\nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) = 0$ separates points where the function increases from those where it decreases. For pseudoconvex functions, the condition $\nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) \geq 0$ implies $f(\mathbf{x}) \geq f(\bar{\mathbf{x}})$.

7.6.2 Log-Convex Functions

A function $f : S \rightarrow \mathbb{R}_{++}$ (where \mathbb{R}_{++} denotes the positive reals) is **log-convex** if $\log f$ is convex. Equivalently, f is log-convex if for all $\mathbf{x}_1, \mathbf{x}_2 \in S$ and $\lambda \in (0, 1)$:

$$f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) \leq f(\mathbf{x}_1)^\lambda f(\mathbf{x}_2)^{1-\lambda}.$$

Log-convex functions arise naturally in many applications, including probability theory (where many density functions are log-concave) and combinatorics.

For more on log-convex functions, see Section 3.5 of Boyd and Vandenberghe.

7.6.3 Other Generalizations

Several other generalizations of convexity exist that are useful in specific contexts:

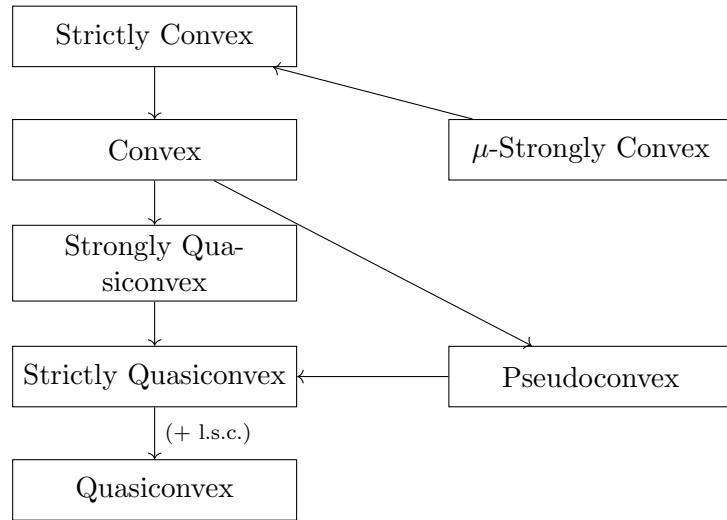
- **Convexity with respect to generalized inequalities:** Extends

convexity to functions taking values in ordered vector spaces. See Section 3.6 of Boyd and Vandenberghe.

- **L -convexity and M -convexity:** Important in discrete convex analysis, these notions extend convexity to functions on integer lattices while preserving polynomial-time solvability of optimization problems.
- **Invexity:** A function f is invex if there exists a vector-valued function $\eta(\mathbf{x}, \mathbf{y})$ such that $f(\mathbf{x}) - f(\mathbf{y}) \geq \nabla f(\mathbf{y})^T \eta(\mathbf{x}, \mathbf{y})$ for all \mathbf{x}, \mathbf{y} . Every differentiable convex function is invex (with $\eta(\mathbf{x}, \mathbf{y}) = \mathbf{x} - \mathbf{y}$), but invexity is more general.

7.7 Summary of Function Classes

The following diagram summarizes the relationships between the various classes of functions discussed in this chapter:



Each function class preserves certain optimization properties:

Function Class	Local \Rightarrow Global	Unique Minimum
Convex	Yes	No
Strictly Convex	Yes	Yes
μ -Strongly Convex	Yes	Yes
Quasiconvex	No	No
Strictly Quasiconvex	Yes	No
Strongly Quasiconvex	Yes	Yes
Pseudoconvex	Yes	No

Part III

Unconstrained Optimization

Chapter 8

Unconstrained Optimization: Conditions and Algorithms

This chapter develops the theory of optimality conditions for unconstrained optimization problems and introduces the main algorithmic frameworks for solving them. We begin with first-order necessary conditions based on the concept of descent directions, then extend to second-order conditions that provide both necessary and sufficient criteria for local optimality. The chapter concludes with an overview of optimization algorithms, focusing on line search methods and their implementation.

Recommended Reading

- Sections 4.1, 8.1, and 8.2 of Bazaraa, Sherali, and Shetty (2006)
- Chapter 2 of Nocedal and Wright

8.1 First-Order Necessary Conditions

We consider the unconstrained optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}),$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a (possibly nonconvex) function. Our goal is to characterize points that could potentially be local minimizers.

8.1.1 Descent Directions

The key concept underlying first-order optimality conditions is that of a descent direction.

Definition 8.1 (Descent Direction). Let $\bar{\mathbf{x}} \in \mathbb{R}^n$ and suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable at $\bar{\mathbf{x}}$. A vector $\mathbf{d} \in \mathbb{R}^n$ is said to be a **descent direction** at $\bar{\mathbf{x}}$ if

$$\nabla f(\bar{\mathbf{x}})^T \mathbf{d} < 0.$$

The terminology “descent direction” is justified by the following fundamental result, which shows that moving in such a direction initially decreases the function value.

Theorem 8.1 (Descent Along a Descent Direction). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be differentiable at \mathbf{x}^* , and let \mathbf{d} be a descent direction at \mathbf{x}^* . Then there exists $\delta > 0$ such that*

$$f(\mathbf{x}^* + \lambda \mathbf{d}) < f(\mathbf{x}^*) \quad \text{for each } \lambda \in (0, \delta).$$

Proof. By the definition of differentiability, we have

$$f(\mathbf{x}^* + \lambda \mathbf{d}) = f(\mathbf{x}^*) + \lambda \nabla f(\mathbf{x}^*)^T \mathbf{d} + o(\lambda),$$

where $o(\lambda)/\lambda \rightarrow 0$ as $\lambda \rightarrow 0$. Since $\nabla f(\mathbf{x}^*)^T \mathbf{d} < 0$, let $\alpha = -\nabla f(\mathbf{x}^*)^T \mathbf{d} > 0$. For sufficiently small $\lambda > 0$, the remainder term $o(\lambda)$ satisfies $|o(\lambda)| < \lambda \alpha / 2$. Therefore,

$$f(\mathbf{x}^* + \lambda \mathbf{d}) < f(\mathbf{x}^*) - \lambda \alpha + \frac{\lambda \alpha}{2} = f(\mathbf{x}^*) - \frac{\lambda \alpha}{2} < f(\mathbf{x}^*).$$

□

The geometric interpretation is clear: the gradient $\nabla f(\mathbf{x}^*)$ points in the direction of steepest ascent, so any direction making an obtuse angle with the gradient (i.e., having negative inner product with it) is a direction of descent.

8.1.2 The First-Order Necessary Condition

Corollary 8.2 (First-Order Necessary Condition for Local Minimum). *Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable at \mathbf{x}^* . If \mathbf{x}^* is a local minimum, then*

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

Proof. We prove the contrapositive. Suppose $\nabla f(\mathbf{x}^*) \neq \mathbf{0}$. Then $\mathbf{d} = -\nabla f(\mathbf{x}^*)$ satisfies

$$\nabla f(\mathbf{x}^*)^T \mathbf{d} = -\|\nabla f(\mathbf{x}^*)\|^2 < 0,$$

so \mathbf{d} is a descent direction. By Theorem 8.1, there exist points arbitrarily close to \mathbf{x}^* with strictly smaller function values, which means \mathbf{x}^* cannot be a local minimum. \square

Definition 8.2 (Stationary Point). A point \mathbf{x}^* satisfying $\nabla f(\mathbf{x}^*) = \mathbf{0}$ is called a **stationary point** (or **critical point**) of f .

Remark 8.1. The first-order necessary condition tells us that every local minimizer must be a stationary point. However, not every stationary point is a local minimizer—it could also be a local maximizer or a saddle point. Second-order conditions help distinguish between these cases.

Example 8.1 (Finding Stationary Points). (a) Consider $f(x) = 3x^2 - 7x$. We have $f'(x) = 6x - 7$, so the stationary point is $x^* = 7/6$. Since $f''(x) = 6 > 0$, this is indeed a local (and global) minimum.

(b) Consider $f(\mathbf{x}) = 2x_1^2 + 3x_1x_2 - 4x_2^2$. The gradient is

$$\nabla f(\mathbf{x}) = \begin{pmatrix} 4x_1 + 3x_2 \\ 3x_1 - 8x_2 \end{pmatrix}.$$

Setting this equal to zero gives the system $4x_1 + 3x_2 = 0$ and $3x_1 - 8x_2 = 0$, which has the unique solution $\mathbf{x}^* = \mathbf{0}$. The Hessian is

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} 4 & 3 \\ 3 & -8 \end{pmatrix},$$

which has eigenvalues approximately 4.49 and -8.49 . Since the Hessian is indefinite, $\mathbf{x}^* = \mathbf{0}$ is a saddle point, not a local minimum.

8.2 Second-Order Necessary and Sufficient Conditions

First-order conditions alone cannot distinguish between local minima, local maxima, and saddle points. Second-order conditions, which involve the Hessian matrix, provide additional discriminating power.

8.2.1 Second-Order Necessary Conditions

Theorem 8.3 (Second-Order Necessary Conditions). *Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice differentiable at \mathbf{x}^* . If \mathbf{x}^* is a local minimum, then:*

- (i) $\nabla f(\mathbf{x}^*) = \mathbf{0}$, and
- (ii) $\nabla^2 f(\mathbf{x}^*)$ is positive semidefinite.

Proof. Condition (i) follows from Corollary 8.2. For condition (ii), suppose for contradiction that $\nabla^2 f(\mathbf{x}^*)$ is not positive semidefinite. Then there exists $\mathbf{d} \in \mathbb{R}^n$ with $\mathbf{d}^T \nabla^2 f(\mathbf{x}^*) \mathbf{d} < 0$.

By Taylor's theorem, for small $\lambda > 0$:

$$f(\mathbf{x}^* + \lambda \mathbf{d}) = f(\mathbf{x}^*) + \lambda \nabla f(\mathbf{x}^*)^T \mathbf{d} + \frac{\lambda^2}{2} \mathbf{d}^T \nabla^2 f(\mathbf{x}^*) \mathbf{d} + o(\lambda^2).$$

Since $\nabla f(\mathbf{x}^*) = \mathbf{0}$, this simplifies to:

$$f(\mathbf{x}^* + \lambda \mathbf{d}) = f(\mathbf{x}^*) + \frac{\lambda^2}{2} \mathbf{d}^T \nabla^2 f(\mathbf{x}^*) \mathbf{d} + o(\lambda^2).$$

For sufficiently small $\lambda > 0$, the term $\frac{\lambda^2}{2} \mathbf{d}^T \nabla^2 f(\mathbf{x}^*) \mathbf{d} < 0$ dominates $o(\lambda^2)$, so $f(\mathbf{x}^* + \lambda \mathbf{d}) < f(\mathbf{x}^*)$. This contradicts \mathbf{x}^* being a local minimum. \square

Remark 8.2. The conditions in Theorem 8.3 are necessary but not sufficient. For example, consider $f(x) = x^3$. At $x^* = 0$, we have $f'(0) = 0$ and $f''(0) = 0$, so both conditions are satisfied. However, $x^* = 0$ is not a local minimum—it is an inflection point.

8.2.2 Second-Order Sufficient Conditions

To obtain sufficient conditions for a local minimum, we need to strengthen the positive semidefiniteness requirement to positive definiteness.

Theorem 8.4 (Second-Order Sufficient Conditions). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be twice differentiable at \mathbf{x}^* . If*

- (i) $\nabla f(\mathbf{x}^*) = \mathbf{0}$, and
- (ii) $\nabla^2 f(\mathbf{x}^*)$ is positive definite,

then \mathbf{x}^ is a strict local minimum.*

Proof. Let $\lambda_{\min} > 0$ be the smallest eigenvalue of $\nabla^2 f(\mathbf{x}^*)$. Then for any $\mathbf{d} \in \mathbb{R}^n$:

$$\mathbf{d}^T \nabla^2 f(\mathbf{x}^*) \mathbf{d} \geq \lambda_{\min} \|\mathbf{d}\|^2.$$

By Taylor's theorem, for \mathbf{x} near \mathbf{x}^* :

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \frac{1}{2} (\mathbf{x} - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*) (\mathbf{x} - \mathbf{x}^*) + o(\|\mathbf{x} - \mathbf{x}^*\|^2).$$

Since $\nabla f(\mathbf{x}^*) = \mathbf{0}$:

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) + \frac{\lambda_{\min}}{2} \|\mathbf{x} - \mathbf{x}^*\|^2 + o(\|\mathbf{x} - \mathbf{x}^*\|^2).$$

For \mathbf{x} sufficiently close to (but different from) \mathbf{x}^* , the quadratic term dominates, giving $f(\mathbf{x}) > f(\mathbf{x}^*)$. \square

Remark 8.3. There is a gap between the necessary and sufficient conditions: when the Hessian is positive semidefinite but not positive definite at a stationary point, we cannot immediately determine whether the point is a local minimum. Higher-order derivatives or other techniques may be needed.

8.2.3 Optimality for Pseudoconvex Functions

For the special class of pseudoconvex functions, first-order conditions are both necessary and sufficient for global optimality.

Theorem 8.5 (Optimality Conditions for Pseudoconvex Functions).

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be pseudoconvex at \mathbf{x}^* . Then \mathbf{x}^* is a global minimum if and only if $\nabla f(\mathbf{x}^*) = \mathbf{0}$.

Recall that a differentiable function f is pseudoconvex if $\nabla f(\mathbf{x}_1)^T(\mathbf{x}_2 - \mathbf{x}_1) \geq 0$ implies $f(\mathbf{x}_2) \geq f(\mathbf{x}_1)$ for all $\mathbf{x}_1, \mathbf{x}_2$ in the domain. This theorem shows that for pseudoconvex functions, finding a stationary point is equivalent to finding a global minimum.

Example 8.2 (Analyzing Critical Points). Consider $f(\mathbf{x}) = \exp(3x_2) - 3x_1 \exp(x_2) + x_1^3$. The gradient is

$$\nabla f(\mathbf{x}) = \begin{pmatrix} -3 \exp(x_2) + 3x_1^2 \\ 3 \exp(3x_2) - 3x_1 \exp(x_2) \end{pmatrix}.$$

Setting $\nabla f(\mathbf{x}) = \mathbf{0}$:

- From the first equation: $x_1^2 = \exp(x_2)$, so $x_1 = \pm \exp(x_2/2)$.
- Substituting into the second equation: $3 \exp(3x_2) = 3x_1 \exp(x_2)$, giving $\exp(2x_2) = x_1$.

Combining these conditions, we need $x_1^2 = \exp(x_2)$ and $x_1 = \exp(2x_2)$. This gives $\exp(4x_2) = \exp(x_2)$, so $4x_2 = x_2$, hence $x_2 = 0$. Then $x_1 = \exp(0) = 1$.

The critical point is $\mathbf{x}^* = (1, 0)^T$. To classify it, we compute the Hessian:

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} 6x_1 & -3 \exp(x_2) \\ -3 \exp(x_2) & 9 \exp(3x_2) - 3x_1 \exp(x_2) \end{pmatrix}.$$

At $\mathbf{x}^* = (1, 0)^T$:

$$\nabla^2 f(\mathbf{x}^*) = \begin{pmatrix} 6 & -3 \\ -3 & 6 \end{pmatrix}.$$

The eigenvalues are $6 + 3 = 9$ and $6 - 3 = 3$, both positive. Hence $\nabla^2 f(\mathbf{x}^*)$ is positive definite, and $\mathbf{x}^* = (1, 0)^T$ is a strict local minimum.

8.3 Overview of Algorithms for Unconstrained Optimization

Having established conditions that characterize local optima, we now turn to algorithms for finding such points. The general algorithmic framework constructs a sequence of iterates $\{\mathbf{x}_k\}_{k=1}^{\infty}$ starting from an initial guess $\mathbf{x}_1 \in \mathbb{R}^n$. The algorithm terminates when no further progress can be made or when an approximate solution has been found.

8.3.1 Two Main Algorithmic Frameworks

There are two principal approaches to unconstrained optimization:

1. **Line Search Methods:** Given the current iterate $\mathbf{x}_k \in \mathbb{R}^n$ and a search direction $\mathbf{d}_k \in \mathbb{R}^n$, these methods move from \mathbf{x}_k in direction \mathbf{d}_k to find a point with lower function value:

$$\lambda_k^* \in \operatorname{argmin}_{\lambda_k \geq 0} f(\mathbf{x}_k + \lambda_k \mathbf{d}_k). \quad (\text{Line search subproblem})$$

The next iterate is then set to $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k^* \mathbf{d}_k$.

2. **Trust Region Methods:** Given the current iterate $\mathbf{x}_k \in \mathbb{R}^n$, these methods construct a local model m_k of f near \mathbf{x}_k and solve:

$$\mathbf{p}_k^* \in \operatorname{argmin}_{\mathbf{p}_k} \left\{ m_k(\mathbf{x}_k + \mathbf{p}_k) : \mathbf{x}_k + \mathbf{p}_k \text{ in trust region} \right\}. \quad (\text{Trust region subproblem})$$

The actual function value $f(\mathbf{x}_k + \mathbf{p}_k^*)$ is then computed to decide whether to accept the step, i.e., whether to set $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k^*$.

The key difference is that line search methods first choose a direction and then determine how far to move, while trust region methods determine a region in which the model is trusted and then find the best step within that region.

8.3.2 The Line Search Approach

In line search methods, each iteration involves two decisions:

1. **Choose a search direction \mathbf{d}_k .** Common choices include:

- Steepest descent: $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$
- Newton direction: $\mathbf{d}_k = -[\nabla^2 f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k)$

- Quasi-Newton directions
- Conjugate gradient directions

2. **Determine the step length** λ_k by (approximately) solving the line search subproblem.

For convergence, the search direction \mathbf{d}_k should typically be a descent direction at \mathbf{x}_k , ensuring that the function value can be decreased by moving in that direction.

8.4 Line Search Methods

The line search subproblem is a one-dimensional optimization problem in the step length λ :

$$\min_{a \leq \lambda \leq b} \theta(\lambda),$$

where $\theta(\lambda) := f(\mathbf{x}_k + \lambda \mathbf{d}_k)$. The interval $[a, b]$ constrains the step length, typically with $a = 0$.

Definition 8.3 (Interval of Uncertainty). The **interval of uncertainty** is the smallest known interval guaranteed to contain the optimal solution. Initially, this is $[a, b]$, and it shrinks as the algorithm progresses.

Remark 8.4. Exact solution of the line search subproblem may be expensive and is often unnecessary. In practice, approximate solutions that provide sufficient decrease in the objective function are used. However, understanding exact line search methods provides important foundations.

8.4.1 Line Search for Strictly Quasiconvex Objectives

When the objective function θ of the line search subproblem is strictly quasiconvex, efficient algorithms can exploit this structure.

Theorem 8.6 (Interval Reduction for Strictly Quasiconvex Functions).

Let $\theta : \mathbb{R} \rightarrow \mathbb{R}$ be a strictly quasiconvex function over the interval $[a, b]$.

Let $\lambda, \mu \in [a, b]$ with $\lambda < \mu$.

(i) If $\theta(\lambda) > \theta(\mu)$, then $\theta(z) \geq \theta(\mu)$ for all $z \in [\lambda, \mu]$.

(ii) If $\theta(\lambda) \leq \theta(\mu)$, then $\theta(z) \geq \theta(\lambda)$ for all $z \in (\mu, b]$.

This theorem provides the basis for interval reduction methods:

- If $\theta(\lambda) > \theta(\mu)$, the minimum cannot lie in $[a, \lambda]$, so the new interval of uncertainty is $[\lambda, b]$.
- If $\theta(\lambda) \leq \theta(\mu)$, the minimum cannot lie in $(\mu, b]$, so the new interval of uncertainty is $[a, \mu]$.

Several classical methods exploit this structure:

Uniform Search

The **uniform search** (or **simultaneous search**) method divides the interval $[a_1, b_1]$ into equal subintervals using grid points $a_1 + k\delta$ for $k = 1, \dots, n$, where $b_1 = a_1 + (n + 1)\delta$. The function θ is evaluated at all n grid points simultaneously.

Let $\hat{\lambda}$ be a grid point with the smallest function value. If θ is strictly quasiconvex, the minimum lies in the interval $[\hat{\lambda} - \delta, \hat{\lambda} + \delta]$.

Dichotomous Search

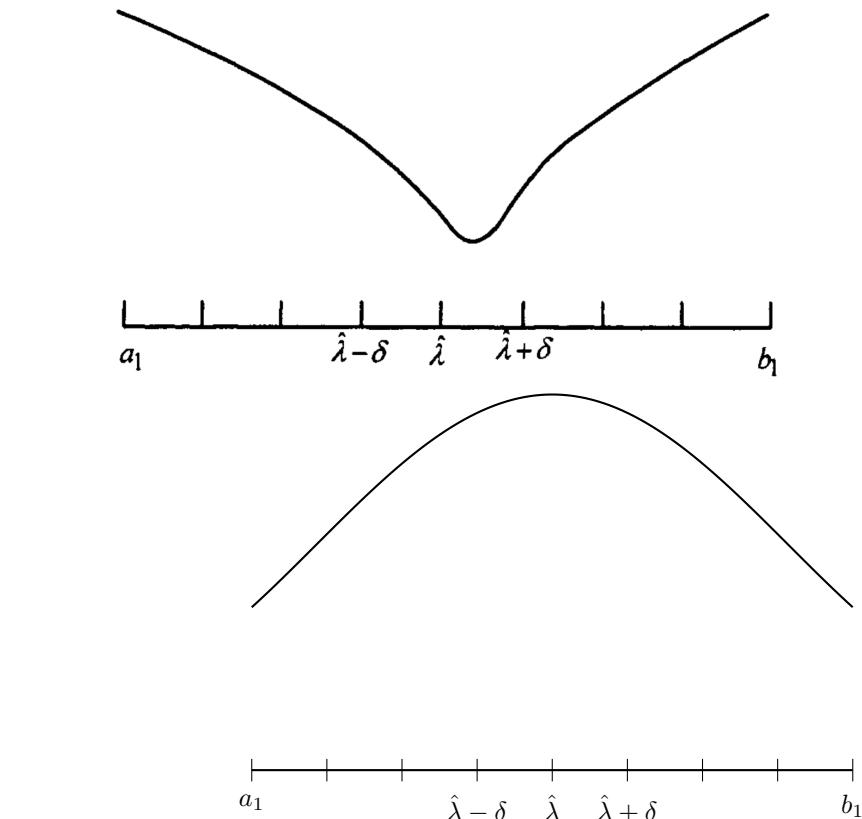
The **dichotomous search** method iteratively bisects the interval of uncertainty:

1. **Initialize:** Set interval of uncertainty $[a_1, b_1]$, tolerance $\epsilon > 0$, and maximum allowable final interval length $l > 0$. Set $k = 1$.
2. **Termination check:** If $b_k - a_k < l$, stop.
3. **Select test points:**

$$\lambda_k = \frac{a_k + b_k}{2} - \epsilon, \quad \mu_k = \frac{a_k + b_k}{2} + \epsilon.$$

4. **Update interval:**

- If $\theta(\lambda_k) < \theta(\mu_k)$, set $a_{k+1} = a_k$ and $b_{k+1} = \mu_k$.
 - Otherwise, set $a_{k+1} = \lambda_k$ and $b_{k+1} = b_k$.
5. Set $k \leftarrow k + 1$ and go to Step 2.



TikZ version:

Figure 8.1: Illustration of the uniform search method. The interval $[a_1, b_1]$ is divided into equal subintervals, and the function is evaluated at each grid point to identify the subinterval containing the minimum.

The length of the interval of uncertainty at the beginning of iteration $k + 1$ is:

$$b_{k+1} - a_{k+1} = \frac{1}{2^k} (b_1 - a_1) + 2\epsilon \left(1 - \frac{1}{2^k}\right).$$

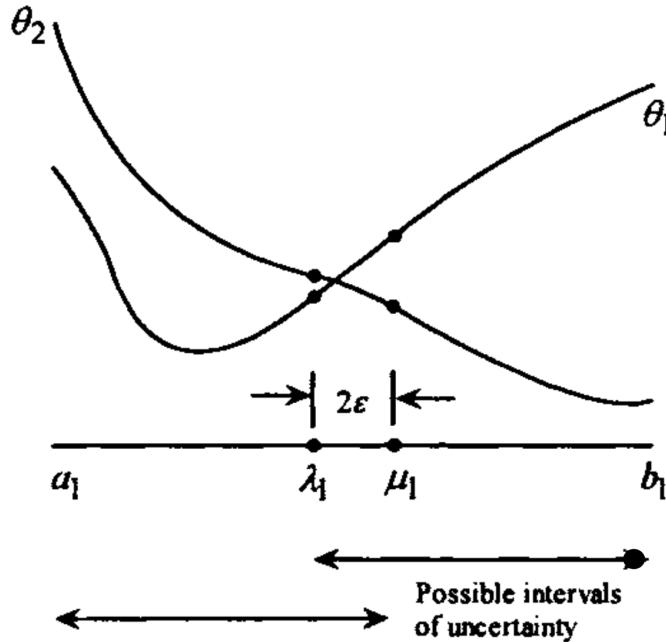


Figure 8.2: Illustration of the dichotomous search method. At each iteration, two test points are placed near the midpoint of the current interval, and the interval is reduced based on function value comparisons.

Golden Section Search

The **golden section search** improves upon dichotomous search by choosing test points so that one point from iteration k can be reused in iteration $k + 1$, reducing the number of function evaluations.

1. **Initialize:** Set interval of uncertainty $[a_1, b_1]$, maximum allowable final interval length $l > 0$. Let $\alpha = \frac{\sqrt{5}-1}{2} \approx 0.618$ (the golden ratio). Set:

$$\lambda_1 = \alpha a_1 + (1 - \alpha)b_1, \quad \mu_1 = (1 - \alpha)a_1 + \alpha b_1.$$

Set $k = 1$.

2. **Termination check:** If $b_k - a_k < l$, stop.

3. **Evaluate and update:**

- If $\theta(\lambda_k) > \theta(\mu_k)$: Set $a_{k+1} = \lambda_k$, $b_{k+1} = b_k$, $\lambda_{k+1} = \mu_k$, and $\mu_{k+1} = \alpha b_{k+1} + (1 - \alpha)a_{k+1}$.
- If $\theta(\lambda_k) \leq \theta(\mu_k)$: Set $a_{k+1} = a_k$, $b_{k+1} = \mu_k$, $\mu_{k+1} = \lambda_k$, and $\lambda_{k+1} = \alpha a_{k+1} + (1 - \alpha)b_{k+1}$.

4. Set $k \leftarrow k + 1$ and go to Step 2.

The key property is that $b_k - \lambda_k = \mu_k - a_k$ for each k , and either $\lambda_{k+1} = \mu_k$ or $\mu_{k+1} = \lambda_k$. This means only one new function evaluation is required at each iteration $k > 1$ (rather than two as in dichotomous search).

The interval is reduced by factor $\alpha \approx 0.618$ at each iteration.

Fibonacci Search

The **Fibonacci search** method achieves optimal reduction of the interval of uncertainty for a given number of function evaluations, using the Fibonacci sequence.

Definition 8.4 (Fibonacci Sequence). The Fibonacci sequence $\{F_i\}$ is defined by:

$$F_{i+1} = F_i + F_{i-1}, \quad i = 1, 2, \dots,$$

with $F_0 = F_1 = 1$. The sequence begins: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Let n be the total budget of function evaluations. Like golden section search, Fibonacci search requires only one function evaluation per iteration after the first. The test points are:

$$\lambda_k = \alpha_k a_k + (1 - \alpha_k) b_k, \quad \mu_k = (1 - \alpha_k) a_k + \alpha_k b_k,$$

where $\alpha_k = F_{n-k}/F_{n-k+1}$ for $k \geq 1$.

The interval of uncertainty reduces by factor α_k at iteration k . Asymptotically, Fibonacci search has similar performance to golden section search, but it is optimal for a fixed budget of function evaluations.

Example 8.3 (Comparing Search Methods). Consider minimizing $\theta(\lambda) = \lambda^2 + 2\lambda$ over $[-3, 5]$. This function is strictly convex (hence strictly quasiconvex) with minimum at $\lambda^* = -1$.

For the golden section method with initial interval $[a_1, b_1] = [-3, 5]$:

- $\lambda_1 = 0.618(-3) + 0.382(5) = 0.056$
- $\mu_1 = 0.382(-3) + 0.618(5) = 1.944$
- $\theta(\lambda_1) = 0.115, \theta(\mu_1) = 7.67$
- Since $\theta(\lambda_1) < \theta(\mu_1)$, new interval is $[-3, 1.944]$

The interval shrinks by approximately 38.2% at each iteration, converging to the optimum.

8.4.2 Line Search Using Derivatives

When derivative information is available, more efficient line search methods can be employed. These methods typically achieve faster convergence than derivative-free methods.

Steepest Descent for Line Search

Given the current iterate λ_k in the one-dimensional line search problem, the **steepest descent** update is:

$$\lambda_{k+1} = \lambda_k - \alpha_k \theta'(\lambda_k),$$

where $\alpha_k > 0$ is a small step size.

Termination occurs when $|\lambda_{k+1} - \lambda_k| < \epsilon$ or $|\theta'(\lambda_k)| < \epsilon$ for a prespecified tolerance $\epsilon > 0$.

Newton's Method for Line Search

Newton's method uses second-order information to achieve faster local convergence.

Assume θ is twice differentiable with $\theta''(\lambda_k) \neq 0$. The method constructs a quadratic approximation of θ at λ_k :

$$q(\lambda) = \theta(\lambda_k) + \theta'(\lambda_k)(\lambda - \lambda_k) + \frac{1}{2}\theta''(\lambda_k)(\lambda - \lambda_k)^2.$$

Setting $q'(\lambda_{k+1}) = 0$ gives the Newton update:

$$\lambda_{k+1} = \lambda_k - \frac{\theta'(\lambda_k)}{\theta''(\lambda_k)}.$$

Termination occurs when $|\lambda_{k+1} - \lambda_k| < \epsilon$ or $|\theta'(\lambda_k)| < \epsilon$.

Remark 8.5 (Convergence of Newton's Method). Newton's method exhibits quadratic convergence near a solution where $\theta''(\lambda^*) \neq 0$. However, it may not converge starting from an arbitrary initial guess. In particular:

- If $\theta''(\lambda_k) = 0$, the method is undefined.
- The method may diverge or cycle if started far from the optimum.
- Newton's method can converge to a maximum or saddle point rather than a minimum.

Practical implementations often combine Newton's method with safeguards such as line search in the Newton direction or trust region modifications.

Example 8.4 (Newton's Method Behavior). Consider the function:

$$\theta(\lambda) = \begin{cases} 4\lambda^3 - 3\lambda^4 & \text{if } \lambda \geq 0 \\ 4\lambda^3 + 3\lambda^4 & \text{if } \lambda < 0 \end{cases}$$

This function has a local minimum at $\lambda = 0$ and local maxima at $\lambda = \pm 1$. Starting from $\lambda_1 = 2$:

- Newton's method converges rapidly to the local minimum at $\lambda = 0$.

However, starting from $\lambda_1 = 0.9$ (near the local maximum):

- Newton's method may diverge or converge to an unexpected point.

This illustrates the sensitivity of Newton's method to the initial guess.

8.4.3 Comparison of Line Search Methods

The following tables provide a comparison of the line search methods discussed in this chapter, summarizing their key properties and computational requirements.

Table 8.1 Summary of Computations for the Golden Section Method

Iteration k	a_k	b_k	λ_k	μ_k	$\theta(\lambda_k)$	$\theta(\mu_k)$
1	-3.000	5.000	0.056	1.944	0.115*	7.667*
2	-3.000	1.944	-1.112	0.056	-0.987*	0.115
3	-3.000	0.056	-1.832	-1.112	-0.308*	-0.987
4	-1.832	0.056	-1.112	-0.664	-0.987	-0.887*
5	-1.832	-0.664	-1.384	-1.112	-0.853*	-0.987
6	-1.384	-0.664	-1.112	-0.936	-0.987	-0.996*
7	-1.112	-0.664	-0.936	-0.840	-0.996	-0.974*
8	-1.112	-0.840	-1.016	-0.936	-1.000*	-0.996
9	-1.112	-0.936				

Figure 8.3: Comparison of derivative-free line search methods: uniform search, dichotomous search, golden section search, and Fibonacci search.

Table 8.2 Summary of Computations for the Fibonacci Search Method

Iteration k	a_k	b_k	λ_k	μ_k	$\theta(\lambda_k)$	$\theta(\mu_k)$
1	-3.000000	5.000000	0.054545	1.945454	0.112065*	7.675699*
2	-3.000000	1.945454	-1.109091	0.054545	-0.988099*	0.112065
3	-3.000000	0.054545	-1.836363	-1.109091	-0.300497*	-0.988099
4	-1.836363	0.054545	-1.109091	-0.672727	-0.988099	-0.892892*
5	-1.836363	-0.672727	-1.399999	-1.109091	-0.840001*	-0.988099
6	-1.399999	-0.672727	-1.109091	-0.963636	-0.988099	-0.998677*
7	-1.109091	-0.672727	-0.963636	-0.818182	-0.998677	-0.966942*
8	-1.109091	-0.818182	-0.963636	-0.963636	-0.998677	-0.998677
9	-1.109091	-0.963636	-0.963636	-0.953636	-0.998677	-0.997850*

Figure 8.4: Iteration counts and interval reduction for derivative-free line search methods.

Table 8.4 Summary of Computations for Newton's Method Starting from $\lambda_1 = 0.4$

Iteration k	λ_k	$\theta'(\lambda_k)$	$\theta''(\lambda_k)$	λ_{k+1}
1	0.400000	1.152000	3.840000	0.100000
2	0.100000	0.108000	2.040000	0.047059
3	0.047059	0.025324	1.049692	0.022934
4	0.022934	0.006167	0.531481	0.011331
5	0.11331	0.001523	0.267322	0.005634
6	0.005634	0.000379	0.134073	0.002807

Figure 8.5: Comparison of derivative-based line search methods: steepest descent and Newton's method.

Table 8.5 Summary of Computations for Newton's Method Starting from $\lambda_1 = 0.6$

Iteration k	λ_k	$\theta'(\lambda_k)$	$\theta''(\lambda_k)$	λ_{k+1}
1	0.600	1.728	1.440	-0.600
2	-0.600	1.728	-1.440	0.600
3	0.600	1.728	1.440	-0.600
4	-0.600	1.728	-1.440	0.600

Figure 8.6: Summary of convergence properties for line search algorithms.

8.5 Summary

This chapter established the theoretical foundations and algorithmic tools for unconstrained optimization:

1. **First-Order Necessary Conditions:** At any local minimum \mathbf{x}^* of a differentiable function, the gradient must vanish: $\nabla f(\mathbf{x}^*) = \mathbf{0}$. This follows from the concept of descent directions.
2. **Second-Order Conditions:**
 - **Necessary:** At a local minimum, the Hessian must be positive semidefinite.
 - **Sufficient:** A stationary point with positive definite Hessian is a strict local minimum.
3. **Algorithmic Frameworks:**
 - **Line search methods:** Choose a direction, then determine how far to move.
 - **Trust region methods:** Determine a trusted region, then find the best step within it.
4. **Line Search Methods:**
 - For strictly quasiconvex objectives: uniform search, dichotomous search, golden section search, Fibonacci search.
 - Using derivatives: steepest descent, Newton's method.

Subsequent chapters will develop specific optimization algorithms—including steepest descent, Newton's method, and conjugate gradient methods—that build upon these foundations.

Chapter 9

The Steepest Descent Method

This chapter introduces the steepest descent method (also known as gradient descent), one of the most fundamental algorithms for unconstrained optimization. We begin by defining descent directions and establishing that the negative gradient provides the direction of steepest descent. We then present the algorithm with exact line search and analyze its convergence properties. The chapter continues with a detailed treatment of the method applied to convex quadratic problems, where we derive closed-form expressions for the step size and establish global convergence. Finally, we discuss inexact line search methods, including the Armijo condition and the Wolfe conditions, which provide practical alternatives to exact line search while maintaining convergence guarantees.

Recommended Reading

- Section 8.6 of Bazaraa, Sherali, and Shetty (2006)
- Sections 3.1–3.3 of Nocedal and Wright
- **Supplementary (Convergence Rate Analysis):** Chapter 3 of Wright and Recht (2022); Sections 3.1–3.4 of Bubeck (2015)

9.1 Introduction to the Steepest Descent Method

Consider the unconstrained (possibly nonconvex) optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}). \quad (9.1)$$

The steepest descent method is an iterative algorithm that moves from one point to another by following the direction in which the objective function decreases most rapidly. To formalize this idea, we first introduce the concept of a descent direction.

Definition 9.1 (Descent Direction). A vector $\mathbf{d}_k \in \mathbb{R}^n$ is called a **descent direction** of a function f at a point \mathbf{x}_k if there exists $\delta > 0$ such that

$$f(\mathbf{x}_k + \lambda \mathbf{d}_k) < f(\mathbf{x}_k) \quad \text{for all } \lambda \in (0, \delta). \quad (9.2)$$

In Chapter ??, we established that the directional derivative $f'(\mathbf{x}_k; \mathbf{d}_k) = \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$ measures the rate of change of f at \mathbf{x}_k in the direction \mathbf{d}_k . This leads to the following characterization.

Lemma 9.1 (Characterization of Descent Directions). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be differentiable at \mathbf{x}_k . A direction \mathbf{d}_k is a descent direction if and only if*

$$f'(\mathbf{x}_k; \mathbf{d}_k) = \nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0. \quad (9.3)$$

The directional derivative $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k$ represents the rate of change of f at \mathbf{x}_k in the direction \mathbf{d}_k . When this quantity is negative, the function decreases as we move from \mathbf{x}_k in the direction \mathbf{d}_k .

A natural question arises: among all possible descent directions, which one provides the greatest decrease? The following lemma answers this question.

Lemma 9.2 (Direction of Steepest Descent). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be differentiable at \mathbf{x}_k with $\nabla f(\mathbf{x}_k) \neq \mathbf{0}$. Then the optimal solution of the problem*

$$\min \{ \nabla f(\mathbf{x}_k)^T \mathbf{d} : \|\mathbf{d}\|_2 \leq 1 \} \quad (9.4)$$

is given by

$$\mathbf{d}^* = -\frac{\nabla f(\mathbf{x}_k)}{\|\nabla f(\mathbf{x}_k)\|_2}, \quad (9.5)$$

which is called the **direction of steepest descent** of f at \mathbf{x}_k .

Proof. By the Cauchy-Schwarz inequality, for any \mathbf{d} with $\|\mathbf{d}\|_2 \leq 1$:

$$\nabla f(\mathbf{x}_k)^T \mathbf{d} \geq -\|\nabla f(\mathbf{x}_k)\|_2 \cdot \|\mathbf{d}\|_2 \geq -\|\nabla f(\mathbf{x}_k)\|_2.$$

Equality holds when $\mathbf{d} = -\nabla f(\mathbf{x}_k)/\|\nabla f(\mathbf{x}_k)\|_2$. \square

Remark 9.1 (Scaling Invariance). In practice, we often use $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$ (without normalization) as the descent direction, since the step size λ_k determined by the line search will compensate for the scaling. This simplifies the algorithm without affecting convergence properties.

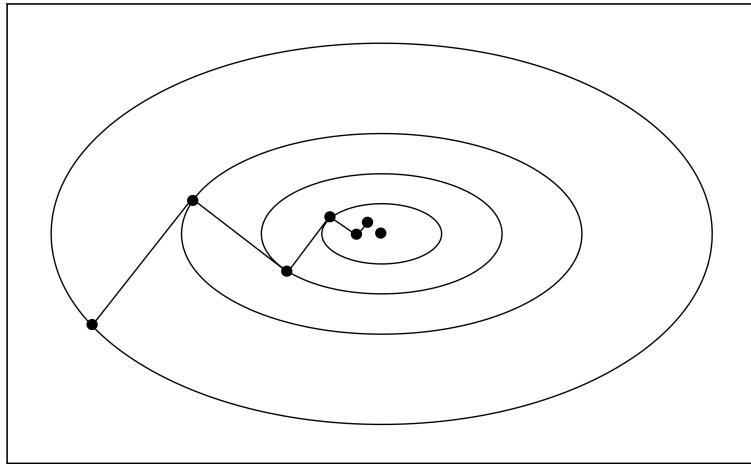


Figure 3.7 Steepest descent steps.

Figure 9.1: Illustration of the gradient descent method. The algorithm iteratively moves from the current point in the direction of the negative gradient, which is the direction of steepest descent. The zigzag pattern is characteristic of gradient descent on problems with elongated level sets.

9.2 Exact Line Search

Having established that $-\nabla f(\mathbf{x}_k)$ is the direction of steepest descent, we now present the complete algorithm. The steepest descent method combines the

choice of direction with a **line search** to determine how far to move along that direction.

9.2.1 Algorithm Description

The steepest descent method with exact line search proceeds as follows:

Algorithm: Steepest Descent with Exact Line Search

Initialization: Choose a termination tolerance $\epsilon > 0$ and a starting point $\mathbf{x}_1 \in \mathbb{R}^n$. Set $k = 1$.

1. **Check termination:** If $\|\nabla f(\mathbf{x}_k)\|_2 < \epsilon$, stop and return \mathbf{x}_k .
2. **Compute descent direction:** Set $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$.
3. **Line search:** Find the step size λ_k by solving

$$\lambda_k \in \operatorname{argmin}_{\lambda \geq 0} f(\mathbf{x}_k + \lambda \mathbf{d}_k). \quad (9.6)$$

4. **Update:** Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$, $k \leftarrow k + 1$, and go to Step 1.

The line search subproblem (9.6) is a one-dimensional optimization problem. Define the univariate function

$$\theta_k(\lambda) \equiv f(\mathbf{x}_k + \lambda \mathbf{d}_k). \quad (9.7)$$

The exact line search finds a global minimizer of θ_k over $\lambda \geq 0$.

Example 9.1 (Minimizing a Simple Quadratic). Consider minimizing $f(\mathbf{x}) = x_1^2 + x_2^2$ starting from $\mathbf{x}_1 = (1, 2)^T$.

Iteration 1:

- Gradient: $\nabla f(\mathbf{x}_1) = (2x_1, 2x_2)^T|_{\mathbf{x}_1} = (2, 4)^T$
- Descent direction: $\mathbf{d}_1 = -\nabla f(\mathbf{x}_1) = (-2, -4)^T$
- Line search function:

$$\theta_1(\lambda) = (1 - 2\lambda)^2 + (2 - 4\lambda)^2 = 5 - 20\lambda + 20\lambda^2$$

- Setting $\theta'_1(\lambda) = -20 + 40\lambda = 0$ gives $\lambda_1 = 1/2$
- Update: $\mathbf{x}_2 = (1, 2)^T + \frac{1}{2}(-2, -4)^T = (0, 0)^T$

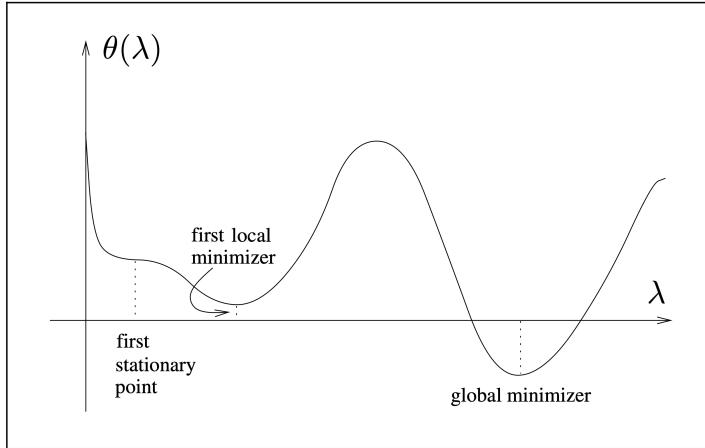


Figure 3.1 The ideal step length is the global minimizer.

Figure 9.2: Illustration of line search. Given a descent direction \mathbf{d}_k , the line search determines the step size λ_k by minimizing the objective function along the ray $\mathbf{x}_k + \lambda \mathbf{d}_k$ for $\lambda \geq 0$.

In this case, the steepest descent method converges to the global minimizer $\mathbf{x}^* = (0, 0)^T$ in a single iteration.

Example 9.2 (A More Challenging Problem). Consider minimizing $f(\mathbf{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$.

This problem illustrates that even for relatively simple functions, the steepest descent method may require many iterations to converge. The zigzag behavior of the algorithm becomes apparent when the level sets of the objective function are highly elongated.

9.2.2 Convergence Analysis

We now establish the fundamental convergence property of the steepest descent method with exact line search.

Theorem 9.3 (Convergence to a Stationary Point). *Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable. If the sequence $\{\mathbf{x}_k : k \geq 1\}$ generated by the steepest descent method converges to a point \mathbf{x}^* , then $\nabla f(\mathbf{x}^*) =$*

0.

Proof. Let $\theta_k(\lambda) \equiv f(\mathbf{x}_k + \lambda \mathbf{d}_k)$ where $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$. Since λ_k minimizes $\theta_k(\lambda)$ over $\lambda \geq 0$, the first-order necessary optimality condition implies

$$\theta'_k(\lambda_k) = 0.$$

Using the chain rule:

$$\theta'_k(\lambda_k) = \frac{d}{d\lambda} f(\mathbf{x}_k + \lambda \mathbf{d}_k) \Big|_{\lambda=\lambda_k} = \nabla f(\mathbf{x}_k + \lambda_k \mathbf{d}_k)^T \mathbf{d}_k.$$

Since $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$ and $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$, we have

$$\theta'_k(\lambda_k) = \nabla f(\mathbf{x}_{k+1})^T (-\nabla f(\mathbf{x}_k)) = -\nabla f(\mathbf{x}_{k+1})^T \nabla f(\mathbf{x}_k) = 0.$$

Therefore,

$$\nabla f(\mathbf{x}_{k+1})^T \nabla f(\mathbf{x}_k) = 0 \quad \text{for all } k \geq 1. \quad (9.8)$$

Taking the limit as $k \rightarrow \infty$ and using the continuity of ∇f :

$$\|\nabla f(\mathbf{x}^*)\|_2^2 = \nabla f(\mathbf{x}^*)^T \nabla f(\mathbf{x}^*) = \lim_{k \rightarrow \infty} \nabla f(\mathbf{x}_{k+1})^T \nabla f(\mathbf{x}_k) = 0.$$

Therefore, $\nabla f(\mathbf{x}^*) = \mathbf{0}$. □

Remark 9.2 (Orthogonality of Consecutive Gradients). The proof reveals an important property: equation (9.8) shows that the gradients at consecutive iterates are orthogonal. This means that

$$\nabla f(\mathbf{x}_{k+1})^T \nabla f(\mathbf{x}_k) = 0 \quad \text{for all } k \geq 1.$$

Since the negative gradient represents the direction of movement at each iteration, this implies that the **directions traversed in two consecutive steps are orthogonal** as well. Indeed:

$$\begin{aligned} (\mathbf{x}_{k+1} - \mathbf{x}_k)^T (\mathbf{x}_{k+2} - \mathbf{x}_{k+1}) &= (\lambda_k \mathbf{d}_k)^T (\lambda_{k+1} \mathbf{d}_{k+1}) \\ &= \lambda_k \lambda_{k+1} (-\nabla f(\mathbf{x}_k))^T (-\nabla f(\mathbf{x}_{k+1})) \\ &= \lambda_k \lambda_{k+1} \nabla f(\mathbf{x}_k)^T \nabla f(\mathbf{x}_{k+1}) = 0. \end{aligned}$$

This orthogonality leads to the characteristic “zigzag” pattern observed when the steepest descent method is applied to problems with elongated level sets.

9.3 Application to Convex Quadratic Problems

We now specialize the steepest descent method to convex quadratic functions, where we can derive closed-form expressions and provide a complete convergence analysis.

Consider a convex quadratic function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ defined by

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T \mathbf{x}, \quad (9.9)$$

where $Q \in \mathbb{R}^{n \times n}$ is a symmetric **positive definite** matrix and $\mathbf{c} \in \mathbb{R}^n$.

For this function:

$$\nabla f(\mathbf{x}) = Q\mathbf{x} + \mathbf{c}, \quad H(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = Q. \quad (9.10)$$

9.3.1 Closed-Form Step Size

At iteration k , the steepest descent method computes

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \nabla f(\mathbf{x}_k),$$

where $\lambda_k \in \operatorname{argmin}_{\lambda \geq 0} \theta_k(\lambda)$ and

$$\theta_k(\lambda) = f(\mathbf{x}_k - \lambda \nabla f(\mathbf{x}_k)).$$

For the convex quadratic function (9.9), we have:

$$\begin{aligned} \theta_k(\lambda) &= \frac{1}{2}(\mathbf{x}_k - \lambda \nabla f(\mathbf{x}_k))^T Q(\mathbf{x}_k - \lambda \nabla f(\mathbf{x}_k)) + \mathbf{c}^T (\mathbf{x}_k - \lambda \nabla f(\mathbf{x}_k)) \\ &= \lambda^2 \left(\frac{1}{2} \nabla f(\mathbf{x}_k)^T Q \nabla f(\mathbf{x}_k) \right) - \lambda (\nabla f(\mathbf{x}_k)^T \nabla f(\mathbf{x}_k)) + f(\mathbf{x}_k). \end{aligned}$$

Since Q is positive definite, the coefficient of λ^2 is positive, so $\theta_k(\lambda)$ is a convex quadratic function in λ . Its global minimizer is found by setting the derivative to zero:

$$\theta'_k(\lambda) = \lambda \cdot \nabla f(\mathbf{x}_k)^T Q \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_k)^T \nabla f(\mathbf{x}_k) = 0.$$

Solving for λ_k :

$$\lambda_k = \frac{\nabla f(\mathbf{x}_k)^T \nabla f(\mathbf{x}_k)}{\nabla f(\mathbf{x}_k)^T Q \nabla f(\mathbf{x}_k)}. \quad (9.11)$$

Proposition 9.4 (Steepest Descent Iteration for Quadratic Functions). *For the convex quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T \mathbf{x}$ with Q positive definite, each iteration of the steepest descent method takes the form*

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\nabla f(\mathbf{x}_k)^T \nabla f(\mathbf{x}_k)}{\nabla f(\mathbf{x}_k)^T Q \nabla f(\mathbf{x}_k)} \nabla f(\mathbf{x}_k) \quad \text{for } k \geq 1. \quad (9.12)$$

Example 9.3 (Steepest Descent for the Euclidean Norm). Consider $f(\mathbf{x}) = \mathbf{x}^T \mathbf{x} = \|\mathbf{x}\|_2^2 = \sum_{i=1}^n x_i^2$.

This corresponds to (9.9) with $Q = 2I_n$ (where I_n is the $n \times n$ identity matrix) and $\mathbf{c} = \mathbf{0}$. The gradient is $\nabla f(\mathbf{x}) = 2\mathbf{x}$.

Given any starting point $\mathbf{x}_1 \in \mathbb{R}^n$:

$$\begin{aligned} \mathbf{x}_2 &= \mathbf{x}_1 - \frac{\nabla f(\mathbf{x}_1)^T \nabla f(\mathbf{x}_1)}{\nabla f(\mathbf{x}_1)^T Q \nabla f(\mathbf{x}_1)} \nabla f(\mathbf{x}_1) \\ &= \mathbf{x}_1 - \frac{4\mathbf{x}_1^T \mathbf{x}_1}{8\mathbf{x}_1^T \mathbf{x}_1} \cdot 2\mathbf{x}_1 \\ &= \mathbf{x}_1 - \mathbf{x}_1 = \mathbf{0}. \end{aligned}$$

Thus, for this problem, steepest descent reaches the global minimizer $\mathbf{x}^* = \mathbf{0}$ in a single iteration from any starting point!

9.3.2 Global Convergence Analysis

We now establish the global convergence of the steepest descent method for convex quadratic problems.

Definition 9.2 (Global Convergence). A numerical optimization method is said to be **globally convergent** if it converges to a stationary solution starting from any initial point.

For the analysis, consider the convex quadratic function (9.9) with positive definite matrix Q . The unique global minimizer is

$$\mathbf{x}^* = -Q^{-1}\mathbf{c}.$$

For convenience, instead of analyzing $f(\mathbf{x})$ directly, we consider the

equivalent function

$$q(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T Q(\mathbf{x} - \mathbf{x}^*). \quad (9.13)$$

Note that $q(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{2}(\mathbf{x}^*)^T Q \mathbf{x}^*$, so q and f differ only by a constant. Importantly, $q(\mathbf{x}) \geq 0$ for all \mathbf{x} , with $q(\mathbf{x}) = 0$ if and only if $\mathbf{x} = \mathbf{x}^*$.

Let $\nabla_k := \nabla q(\mathbf{x}_k) = Q(\mathbf{x}_k - \mathbf{x}^*)$. The steepest descent iteration for q is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \nabla_k, \quad \text{where } \lambda_k = \frac{\nabla_k^T \nabla_k}{\nabla_k^T Q \nabla_k}.$$

Lemma 9.5 (Recurrence Relation). *The function values satisfy*

$$q(\mathbf{x}_{k+1}) = q(\mathbf{x}_k) \left(1 - \frac{\|\nabla_k\|_2^4}{(\nabla_k^T Q \nabla_k)(\nabla_k^T Q^{-1} \nabla_k)} \right). \quad (9.14)$$

Proof. We have

$$\begin{aligned} q(\mathbf{x}_{k+1}) &= \frac{1}{2}(\mathbf{x}_k - \lambda_k \nabla_k - \mathbf{x}^*)^T Q(\mathbf{x}_k - \lambda_k \nabla_k - \mathbf{x}^*) \\ &= q(\mathbf{x}_k) - \lambda_k \nabla_k^T Q(\mathbf{x}_k - \mathbf{x}^*) + \frac{1}{2} \lambda_k^2 \nabla_k^T Q \nabla_k \\ &= q(\mathbf{x}_k) - \lambda_k \nabla_k^T \nabla_k + \frac{1}{2} \lambda_k^2 \nabla_k^T Q \nabla_k. \end{aligned}$$

Substituting $\lambda_k = \frac{\nabla_k^T \nabla_k}{\nabla_k^T Q \nabla_k}$ and using the fact that $2q(\mathbf{x}_k) = \nabla_k^T Q^{-1} \nabla_k$ (which follows from $\nabla_k = Q(\mathbf{x}_k - \mathbf{x}^*)$):

$$\begin{aligned} q(\mathbf{x}_{k+1}) &= q(\mathbf{x}_k) \left(1 - 2 \frac{\|\nabla_k\|_2^4}{(\nabla_k^T Q \nabla_k)(\nabla_k^T Q^{-1} \nabla_k)} + \frac{\|\nabla_k\|_2^4}{(\nabla_k^T Q \nabla_k)(\nabla_k^T Q^{-1} \nabla_k)} \right) \\ &= q(\mathbf{x}_k) \left(1 - \frac{\|\nabla_k\|_2^4}{(\nabla_k^T Q \nabla_k)(\nabla_k^T Q^{-1} \nabla_k)} \right). \end{aligned}$$

□

Theorem 9.6 (Global Convergence for Convex Quadratic Functions). *Let Q be a symmetric positive definite matrix with smallest eigenvalue $\alpha_{\min}(Q)$ and largest eigenvalue $\alpha_{\max}(Q)$. Then the steepest descent*

method satisfies

$$q(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_k) \left(1 - \frac{\alpha_{\min}(Q)}{\alpha_{\max}(Q)}\right) \quad (9.15)$$

and consequently

$$q(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_1) \left(1 - \frac{\alpha_{\min}(Q)}{\alpha_{\max}(Q)}\right)^k. \quad (9.16)$$

Proof. By Rayleigh's inequality, for any $\mathbf{z} \neq \mathbf{0}$:

$$\alpha_{\min}(Q) \leq \frac{\mathbf{z}^T Q \mathbf{z}}{\mathbf{z}^T \mathbf{z}} \leq \alpha_{\max}(Q).$$

Therefore:

$$\nabla_k^T Q \nabla_k \leq \alpha_{\max}(Q) \|\nabla_k\|_2^2$$

and

$$\nabla_k^T Q^{-1} \nabla_k \leq \alpha_{\max}(Q^{-1}) \|\nabla_k\|_2^2 = \frac{1}{\alpha_{\min}(Q)} \|\nabla_k\|_2^2.$$

Substituting into Lemma 9.5:

$$q(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_k) \left(1 - \frac{\|\nabla_k\|_2^4}{\alpha_{\max}(Q) \|\nabla_k\|_2^2 \cdot \frac{1}{\alpha_{\min}(Q)} \|\nabla_k\|_2^2}\right) = q(\mathbf{x}_k) \left(1 - \frac{\alpha_{\min}(Q)}{\alpha_{\max}(Q)}\right).$$

The bound (9.16) follows by induction. \square

Corollary 9.7 (Convergence to the Minimizer). *Under the assumptions of Theorem 9.6:*

1. $q(\mathbf{x}_k) \rightarrow 0$ as $k \rightarrow \infty$.
2. $\mathbf{x}_k \rightarrow \mathbf{x}^*$ as $k \rightarrow \infty$.
3. The steepest descent method is globally convergent for convex quadratic functions.
4. The rate of convergence is **linear** (also called geometric or exponential).

Remark 9.3 (Condition Number and Convergence Rate). Several important observations follow from Theorem 9.6:

- If $\alpha_{\min}(Q) = \alpha_{\max}(Q)$, then $Q = \alpha I$ for some $\alpha > 0$, and steepest descent converges in **one step**. This corresponds to level sets that are spherical.
- If $\alpha_{\max}(Q) \gg \alpha_{\min}(Q)$, then $1 - \frac{\alpha_{\min}(Q)}{\alpha_{\max}(Q)} \approx 1$, and convergence may be **extremely slow**.
- The ratio
$$\kappa(Q) = \frac{\alpha_{\max}(Q)}{\alpha_{\min}(Q)} = \|Q\|_2 \cdot \|Q^{-1}\|_2 \quad (9.17)$$
is called the **condition number** of Q .
- A matrix with a large condition number is called **ill-conditioned**. This case corresponds to “long, narrow” elliptical level sets, where the steepest descent method exhibits a characteristic zigzag behavior as it searches for the minimizer.

9.4 Convergence Rate Analysis: A Modern Perspective

Modern optimization theory characterizes algorithms by their **convergence rates**—how quickly the error decreases as a function of the iteration count. This perspective, developed extensively in the machine learning literature, provides a unifying framework for comparing and analyzing optimization algorithms.

Recommended Reading

Supplementary Reading: For rigorous proofs and additional results, see Chapter 3 of Wright and Recht (2022) and Sections 3.1–3.4 of Bubeck (2015).

9.4.1 Key Assumptions: Smoothness and Strong Convexity

The convergence rate of gradient descent depends critically on two properties of the objective function.

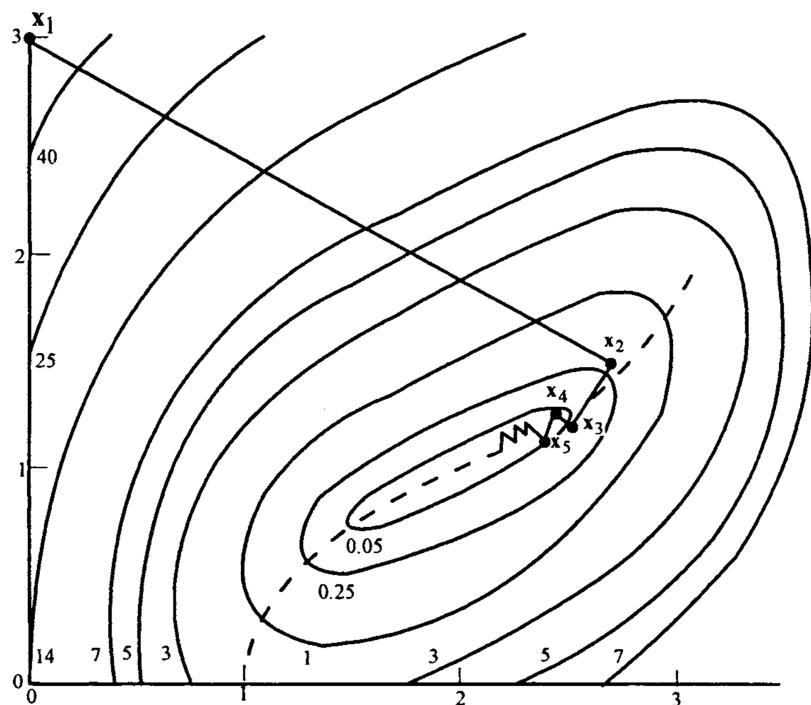


Figure 9.3: Behavior of the steepest descent method on an ill-conditioned problem. The characteristic zigzag pattern occurs when the level sets are highly elongated ellipses, causing the algorithm to make slow progress toward the minimizer.

Definition 9.3 (*L*-Smoothness). A differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is ***L*-smooth** (or has ***L*-Lipschitz continuous gradient**) if there exists a constant $L > 0$ such that

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2 \quad \text{for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \quad (9.18)$$

For twice-differentiable functions, *L*-smoothness is equivalent to $\|\nabla^2 f(\mathbf{x})\|_2 \leq L$ for all \mathbf{x} . In particular, for a quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}$, the smoothness constant is $L = \alpha_{\max}(Q)$.

Definition 9.4 (μ -Strong Convexity). A differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is **μ -strongly convex** if there exists a constant $\mu > 0$ such that

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 \quad \text{for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \quad (9.19)$$

For twice-differentiable functions, μ -strong convexity is equivalent to $\nabla^2 f(\mathbf{x}) \succeq \mu I$ for all \mathbf{x} . For a quadratic with $Q \succ 0$, the strong convexity constant is $\mu = \alpha_{\min}(Q)$.

Remark 9.4 (Condition Number Revisited). The **condition number** of an *L*-smooth and μ -strongly convex function is

$$\kappa = \frac{L}{\mu}.$$

This generalizes the condition number $\kappa(Q) = \frac{\alpha_{\max}(Q)}{\alpha_{\min}(Q)}$ from the quadratic case. A well-conditioned problem has κ close to 1; an ill-conditioned problem has $\kappa \gg 1$.

9.4.2 Convergence Rates for Gradient Descent

We now state the fundamental convergence rate results for gradient descent with a fixed step size $\lambda = 1/L$.

Theorem 9.8 (Convergence for Smooth Convex Functions). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be convex and *L*-smooth. Consider gradient descent with step*

size $\lambda = 1/L$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{L} \nabla f(\mathbf{x}_k).$$

Then for any minimizer \mathbf{x}^* of f :

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{L\|\mathbf{x}_1 - \mathbf{x}^*\|_2^2}{2k} = O\left(\frac{1}{k}\right). \quad (9.20)$$

Proof. The proof relies on two key lemmas about L -smooth functions.

Lemma 1 (Descent Lemma): If f is L -smooth, then for all \mathbf{x}, \mathbf{y} :

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2.$$

Lemma 2 (Convexity): If f is convex, then for all \mathbf{x}, \mathbf{y} :

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}).$$

Step 1: Sufficient decrease per iteration. Applying Lemma 1 with $\mathbf{y} = \mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{L} \nabla f(\mathbf{x}_k)$:

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T(\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ &= f(\mathbf{x}_k) - \frac{1}{L}\|\nabla f(\mathbf{x}_k)\|^2 + \frac{L}{2} \cdot \frac{1}{L^2}\|\nabla f(\mathbf{x}_k)\|^2 \\ &= f(\mathbf{x}_k) - \frac{1}{2L}\|\nabla f(\mathbf{x}_k)\|^2. \end{aligned}$$

Step 2: Relate gradient norm to optimality gap. By convexity (Lemma 2) with $\mathbf{y} = \mathbf{x}^*$:

$$f(\mathbf{x}^*) \geq f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T(\mathbf{x}^* - \mathbf{x}_k).$$

Rearranging: $\nabla f(\mathbf{x}_k)^T(\mathbf{x}_k - \mathbf{x}^*) \leq f(\mathbf{x}_k) - f(\mathbf{x}^*)$.

Step 3: Track progress toward \mathbf{x}^* . Define $\delta_k = f(\mathbf{x}_k) - f(\mathbf{x}^*)$. We have:

$$\begin{aligned} \|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 &= \|\mathbf{x}_k - \frac{1}{L} \nabla f(\mathbf{x}_k) - \mathbf{x}^*\|^2 \\ &= \|\mathbf{x}_k - \mathbf{x}^*\|^2 - \frac{2}{L} \nabla f(\mathbf{x}_k)^T(\mathbf{x}_k - \mathbf{x}^*) + \frac{1}{L^2}\|\nabla f(\mathbf{x}_k)\|^2 \\ &\leq \|\mathbf{x}_k - \mathbf{x}^*\|^2 - \frac{2}{L}\delta_k + \frac{1}{L^2}\|\nabla f(\mathbf{x}_k)\|^2. \end{aligned}$$

From Step 1: $\|\nabla f(\mathbf{x}_k)\|^2 \leq 2L(\delta_k - \delta_{k+1})$. Substituting:

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 \leq \|\mathbf{x}_k - \mathbf{x}^*\|^2 - \frac{2}{L}\delta_k + \frac{2}{L}(\delta_k - \delta_{k+1}) = \|\mathbf{x}_k - \mathbf{x}^*\|^2 - \frac{2}{L}\delta_{k+1}.$$

Step 4: Telescope and conclude. Rearranging: $\delta_{k+1} \leq \frac{L}{2}(\|\mathbf{x}_k - \mathbf{x}^*\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2)$.

Summing from $k = 1$ to $k = K - 1$:

$$\sum_{k=2}^K \delta_k \leq \frac{L}{2} \|\mathbf{x}_1 - \mathbf{x}^*\|^2.$$

Since δ_k is decreasing (from Step 1), $(K - 1)\delta_K \leq \sum_{k=2}^K \delta_k$, giving:

$$\delta_K \leq \frac{L\|\mathbf{x}_1 - \mathbf{x}^*\|^2}{2(K - 1)} \leq \frac{L\|\mathbf{x}_1 - \mathbf{x}^*\|^2}{2K} \quad \text{for } K \geq 2.$$

□

Remark 9.5 (Sublinear Convergence). The $O(1/k)$ rate is called **sub-linear** convergence. To achieve ϵ -accuracy (i.e., $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$), we need $k = O(1/\epsilon)$ iterations. This is relatively slow—halving the error requires doubling the number of iterations.

Theorem 9.9 (Convergence for Smooth Strongly Convex Functions). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be μ -strongly convex and L -smooth. Consider gradient descent with step size $\lambda = 1/L$:*

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{L} \nabla f(\mathbf{x}_k).$$

Then:

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \left(1 - \frac{\mu}{L}\right)^{k-1} (f(\mathbf{x}_1) - f(\mathbf{x}^*)) = \left(1 - \frac{1}{\kappa}\right)^{k-1} (f(\mathbf{x}_1) - f(\mathbf{x}^*)). \quad (9.21)$$

Equivalently, $\|\mathbf{x}_k - \mathbf{x}^*\|_2^2 \leq \left(1 - \frac{1}{\kappa}\right)^{k-1} \|\mathbf{x}_1 - \mathbf{x}^*\|_2^2$.

Proof. The key additional ingredient is the **Polyak-Łojasiewicz (PL) inequality**, which holds for μ -strongly convex functions:

$$\|\nabla f(\mathbf{x})\|^2 \geq 2\mu(f(\mathbf{x}) - f(\mathbf{x}^*)) \quad \text{for all } \mathbf{x}.$$

Derivation of PL inequality: By μ -strong convexity applied at the minimizer \mathbf{x}^* :

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^*\|^2 = f(\mathbf{x}^*) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^*\|^2,$$

where we used $\nabla f(\mathbf{x}^*) = \mathbf{0}$. This gives $\|\mathbf{x} - \mathbf{x}^*\|^2 \leq \frac{2}{\mu}(f(\mathbf{x}) - f(\mathbf{x}^*))$.

By L -smoothness (co-coercivity of the gradient):

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^*)\|^2 \leq L \cdot \nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{x}^*).$$

Since $\nabla f(\mathbf{x}^*) = \mathbf{0}$ and by Cauchy-Schwarz: $\|\nabla f(\mathbf{x})\| \cdot \|\mathbf{x} - \mathbf{x}^*\| \geq \nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{x}^*)$.

A cleaner argument: by convexity, $f(\mathbf{x}) - f(\mathbf{x}^*) \leq \nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{x}^*) \leq \|\nabla f(\mathbf{x})\| \cdot \|\mathbf{x} - \mathbf{x}^*\|$. Combined with the strong convexity bound on $\|\mathbf{x} - \mathbf{x}^*\|$:

$$f(\mathbf{x}) - f(\mathbf{x}^*) \leq \|\nabla f(\mathbf{x})\| \cdot \sqrt{\frac{2}{\mu}(f(\mathbf{x}) - f(\mathbf{x}^*))}.$$

Squaring: $(f(\mathbf{x}) - f(\mathbf{x}^*))^2 \leq \frac{2}{\mu}\|\nabla f(\mathbf{x})\|^2(f(\mathbf{x}) - f(\mathbf{x}^*))$, giving the PL inequality.

Main proof: From the descent lemma (proved in Theorem 9.8):

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{1}{2L}\|\nabla f(\mathbf{x}_k)\|^2.$$

Applying the PL inequality:

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{1}{2L} \cdot 2\mu(f(\mathbf{x}_k) - f(\mathbf{x}^*)) = f(\mathbf{x}_k) - \frac{\mu}{L}(f(\mathbf{x}_k) - f(\mathbf{x}^*)).$$

Subtracting $f(\mathbf{x}^*)$ from both sides:

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) \leq \left(1 - \frac{\mu}{L}\right)(f(\mathbf{x}_k) - f(\mathbf{x}^*)) = \left(1 - \frac{1}{\kappa}\right)(f(\mathbf{x}_k) - f(\mathbf{x}^*)).$$

Applying this recursively:

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \left(1 - \frac{1}{\kappa}\right)^{k-1}(f(\mathbf{x}_1) - f(\mathbf{x}^*)).$$

The bound on $\|\mathbf{x}_k - \mathbf{x}^*\|^2$ follows from strong convexity: $f(\mathbf{x}_k) - f(\mathbf{x}^*) \geq \frac{\mu}{2}\|\mathbf{x}_k - \mathbf{x}^*\|^2$. \square

Remark 9.6 (Linear Convergence). The rate $(1 - \frac{1}{\kappa})^k$ is called **linear** (or **geometric**) convergence. To achieve ϵ -accuracy, we need $k = O(\kappa \log(1/\epsilon))$ iterations. This is much faster than sublinear convergence: each iteration reduces the error by a constant factor.

Remark 9.7 (Comparison with Quadratic Analysis). Theorem 9.9 generalizes our earlier result for quadratics (Theorem 9.6). For the quadratic $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T \mathbf{x}$:

- $L = \alpha_{\max}(Q)$ and $\mu = \alpha_{\min}(Q)$
- The condition number $\kappa = L/\mu = \alpha_{\max}(Q)/\alpha_{\min}(Q)$
- The convergence factor $1 - 1/\kappa = 1 - \alpha_{\min}(Q)/\alpha_{\max}(Q)$

This matches the bound in (9.15) exactly.

9.4.3 Summary of Convergence Rates

The following table summarizes the convergence rates for gradient descent:

Function Class	Rate	Iterations for ϵ -accuracy
Convex, L -smooth	$O(1/k)$	$O(L/\epsilon)$
μ -strongly convex, L -smooth	$O((1 - 1/\kappa)^k)$	$O(\kappa \log(1/\epsilon))$

The key insights are:

1. **Strong convexity dramatically improves convergence:** Moving from convex to strongly convex changes the dependence on accuracy from $O(1/\epsilon)$ to $O(\log(1/\epsilon))$.
2. **Condition number is the key quantity:** For strongly convex problems, the iteration complexity depends on $\kappa = L/\mu$. Ill-conditioned problems require many more iterations.
3. **Fixed step size is simple and effective:** Using $\lambda = 1/L$ (rather than exact line search) achieves the same asymptotic rates with less computation per iteration.

Remark 9.8 (Accelerated Methods). The $O(1/k)$ rate for convex functions and the $O((1 - 1/\kappa)^k)$ rate for strongly convex functions can both be improved. Nesterov's accelerated gradient method achieves:

- $O(1/k^2)$ for convex, L -smooth functions
- $O((1 - 1/\sqrt{\kappa})^k)$ for μ -strongly convex, L -smooth functions

These accelerated methods are covered in Chapter 12.

9.5 Inexact Line Search

In practice, solving the exact line search problem (9.6) can be computationally expensive. Moreover, it is often unnecessary to find the exact minimizer—what matters is that each step achieves sufficient decrease in the objective function while maintaining convergence guarantees.

Inexact line search methods aim to identify a step length λ_k that achieves adequate reduction in f at minimal computational cost while preserving convergence.

9.5.1 The Sufficient Decrease Condition (Armijo's Rule)

The most basic requirement for a step length is that it provides sufficient decrease in the objective function.

Definition 9.5 (Sufficient Decrease Condition / Armijo's Rule). Let $c_1 \in (0, 1)$ be a parameter. The step length λ satisfies the **sufficient decrease condition** (also called **Armijo's rule**) if

$$f(\mathbf{x}_k + \lambda \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \lambda \nabla f(\mathbf{x}_k)^T \mathbf{d}_k. \quad (9.22)$$

The right-hand side of (9.22) defines a linear function of λ :

$$\ell(\lambda) = f(\mathbf{x}_k) + c_1 \lambda \nabla f(\mathbf{x}_k)^T \mathbf{d}_k.$$

Since \mathbf{d}_k is a descent direction, $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$, so $\ell(\lambda)$ is a decreasing line. The Armijo condition requires that the actual function value $f(\mathbf{x}_k + \lambda \mathbf{d}_k)$ lies below this line.

The parameter c_1 controls how much decrease we require. Typical values are $c_1 = 10^{-4}$, which imposes only a mild requirement on the decrease.

Remark 9.9 (Geometric Interpretation). The sufficient decrease condition ensures that the reduction in f is proportional to the step length and the directional derivative. For small λ , the condition is easily satisfied since $f(\mathbf{x}_k + \lambda \mathbf{d}_k) \approx f(\mathbf{x}_k) + \lambda \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$ and $c_1 < 1$.

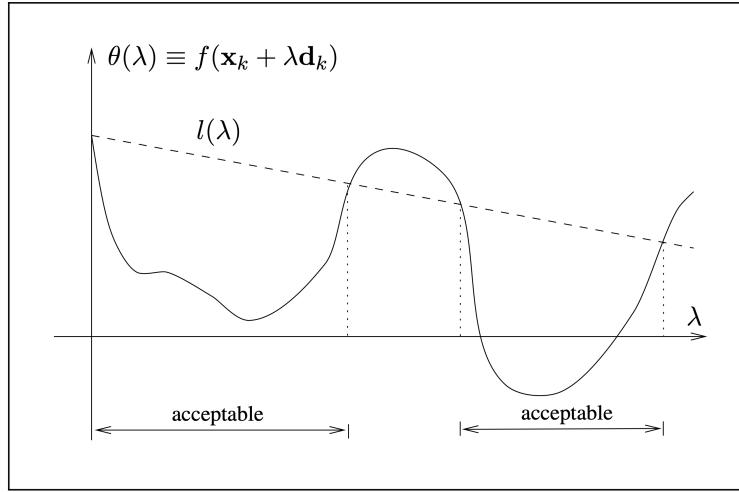


Figure 3.3 Sufficient decrease condition.

Figure 9.4: The Armijo sufficient decrease condition. The step length λ is acceptable if the function value $f(\mathbf{x}_k + \lambda \mathbf{d}_k)$ lies below the line $\ell(\lambda) = f(\mathbf{x}_k) + c_1 \lambda \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$. The acceptable region is shown where the actual function value (curved line) is below the linear bound.

9.5.2 The Wolfe Conditions

The Armijo condition alone is not sufficient to guarantee convergence, as it can be satisfied by arbitrarily small step sizes. To rule out such steps, we add a second condition that prevents the step length from being too small.

Definition 9.6 (Curvature Condition). Let $c_2 \in (c_1, 1)$ be a parameter. The step length λ satisfies the **curvature condition** if

$$\nabla f(\mathbf{x}_k + \lambda \mathbf{d}_k)^T \mathbf{d}_k \geq c_2 \nabla f(\mathbf{x}_k)^T \mathbf{d}_k. \quad (9.23)$$

Since $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$ for a descent direction, the curvature condition

requires that the slope of $\theta_k(\lambda) = f(\mathbf{x}_k + \lambda \mathbf{d}_k)$ at the chosen step is less negative than the initial slope multiplied by c_2 . This ensures that we have moved far enough along the descent direction.

Definition 9.7 (Wolfe Conditions). The step length λ satisfies the **Wolfe conditions** if it satisfies both:

1. The sufficient decrease condition (9.22)
2. The curvature condition (9.23)

with $0 < c_1 < c_2 < 1$.

Remark 9.10 (Typical Parameter Values). A common choice of parameters is $c_1 = 10^{-4}$ and $c_2 = 0.9$. The value $c_1 = 10^{-4}$ imposes a very mild decrease requirement, while $c_2 = 0.9$ allows substantial flexibility in the curvature condition.

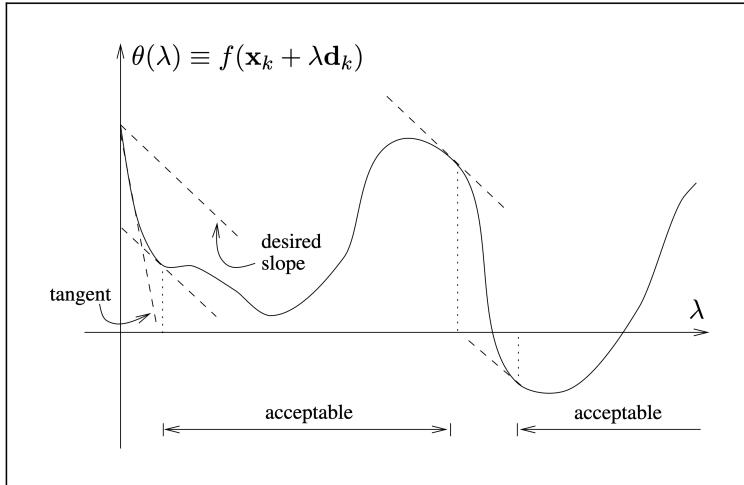


Figure 3.4 The curvature condition.

Figure 9.5: The curvature (Wolfe) condition. The step length λ is acceptable if the slope of the function at $\mathbf{x}_k + \lambda \mathbf{d}_k$ is greater than c_2 times the initial slope. This condition prevents the step size from being too small by requiring that we have moved far enough along the descent direction.

9.5.3 Backtracking Line Search

A simple and widely used method for finding a step length satisfying the Armijo condition is the backtracking line search.

Algorithm: Backtracking Line Search

Input: Descent direction \mathbf{d}_k , current point \mathbf{x}_k .

Parameters: Maximum step length $\bar{\lambda} > 0$, sufficient decrease parameter $c_1 \in (0, 1)$, contraction factor $\rho \in (0, 1)$.

Initialization: Set $\lambda = \bar{\lambda}$.

1. **Check Armijo condition:** If

$$f(\mathbf{x}_k + \lambda \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \lambda \nabla f(\mathbf{x}_k)^T \mathbf{d}_k,$$

then terminate and return λ .

2. **Contract:** Set $\lambda \leftarrow \rho \lambda$ and go to Step 1.

The backtracking algorithm starts with a large step size and repeatedly shrinks it by the factor ρ until the Armijo condition is satisfied. Typical values are $\bar{\lambda} = 1$, $c_1 = 10^{-4}$, and $\rho \in [0.1, 0.5]$.

Remark 9.11 (Termination Guarantee). The backtracking line search is guaranteed to terminate in finite iterations. Since \mathbf{d}_k is a descent direction, we have $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$. For sufficiently small λ , the Taylor expansion gives

$$f(\mathbf{x}_k + \lambda \mathbf{d}_k) \approx f(\mathbf{x}_k) + \lambda \nabla f(\mathbf{x}_k)^T \mathbf{d}_k < f(\mathbf{x}_k) + c_1 \lambda \nabla f(\mathbf{x}_k)^T \mathbf{d}_k,$$

so the Armijo condition will eventually be satisfied.

Remark 9.12 (More Sophisticated Methods). For line search methods that satisfy the full Wolfe conditions (including the curvature condition), more sophisticated algorithms are needed. See Algorithm 3.5 of Nocedal and Wright for an inexact line search algorithm satisfying the *strong* Wolfe conditions.

9.5.4 Numerical Examples

The following tables illustrate the convergence behavior of the steepest descent method on various test problems.

Table 8.4 Summary of Computations for Newton's Method Starting from $\lambda_1 = 0.4$

Iteration k	λ_k	$\theta'(\lambda_k)$	$\theta''(\lambda_k)$	λ_{k+1}
1	0.400000	1.152000	3.840000	0.100000
2	0.100000	0.108000	2.040000	0.047059
3	0.047059	0.025324	1.049692	0.022934
4	0.022934	0.006167	0.531481	0.011331
5	0.11331	0.001523	0.267322	0.005634
6	0.005634	0.000379	0.134073	0.002807

Figure 9.6: Iteration history for the steepest descent method applied to a quadratic function. The table shows the iterate \mathbf{x}_k , function value $f(\mathbf{x}_k)$, gradient norm $\|\nabla f(\mathbf{x}_k)\|$, and step size λ_k at each iteration.

9.6 Summary

The steepest descent method is one of the most fundamental algorithms in optimization. Its key properties are:

- **Simplicity:** The method only requires gradient information and is easy to implement.
- **Global convergence:** For convex quadratic functions, the method is globally convergent, meaning it converges to the minimizer from any starting point.
- **Linear convergence:** The rate of convergence is linear, with the convergence factor depending on the condition number of the Hessian.
- **Sensitivity to conditioning:** For ill-conditioned problems (large condition number), convergence can be very slow due to the zigzag behavior of the iterates.

Table 8.5 Summary of Computations for Newton's Method Starting from $\lambda_1 = 0.6$

Iteration k	λ_k	$\theta'(\lambda_k)$	$\theta''(\lambda_k)$	λ_{k+1}
1	0.600	1.728	1.440	-0.600
2	-0.600	1.728	-1.440	0.600
3	0.600	1.728	1.440	-0.600
4	-0.600	1.728	-1.440	0.600

Figure 9.7: Additional iteration data for the steepest descent method, demonstrating the linear convergence rate and the relationship between the condition number and convergence speed.

Table 8.11 Summary of Computations for the Method of Steepest Descent

Iteration k	\mathbf{x}_k $f(\mathbf{x}_k)$	$\nabla f(\mathbf{x}_k)$	$\ \nabla f(\mathbf{x}_k)\ $	$\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$	λ_k	\mathbf{x}_{k+1}
1	(0.00, 3.00) 52.00	(-44.00, 24.00)	50.12	(44.00, -24.00)	0.062	(2.70, 1.51)
2	(2.70, 1.51) 0.34	(0.73, 1.28)	1.47	(-0.73, -1.28)	0.24	(2.52, 1.20)
3	(2.52, 1.20) 0.09	(0.80, -0.48)	0.93	(-0.80, 0.48)	0.11	(2.43, 1.25)
4	(2.43, 1.25) 0.04	(0.18, 0.28)	0.33	(-0.18, -0.28)	0.31	(2.37, 1.16)
5	(2.37, 1.16) 0.02	(0.30, -0.20)	0.36	(-0.30, 0.20)	0.12	(2.33, 1.18)
6	(2.33, 1.18) 0.01	(0.08, 0.12)	0.14	(-0.08, -0.12)	0.36	(2.30, 1.14)
7	(2.30, 1.14) 0.009	(0.15, -0.08)	0.17	(-0.15, 0.08)	0.13	(2.28, 1.15)
8	(2.28, 1.15) 0.007	(0.05, 0.08)	0.09			

Figure 9.8: Comparison of steepest descent iterations with different line search strategies. The table illustrates how inexact line search methods can achieve comparable convergence with reduced computational cost per iteration.

- **Practical line search:** Inexact line search methods such as back-tracking with Armijo's rule provide efficient alternatives to exact line search while maintaining convergence guarantees.

The limitations of steepest descent, particularly its slow convergence for ill-conditioned problems, motivate the development of more sophisticated methods such as Newton's method and conjugate gradient methods, which will be discussed in subsequent chapters.

Chapter 10

Newton's Method and Quasi-Newton Methods

This chapter introduces Newton's method for unconstrained optimization, a powerful second-order algorithm that exploits curvature information through the Hessian matrix. We begin with the motivation and derivation of Newton's method from the second-order Taylor series approximation, then examine its application to convex quadratic programs and analyze its convergence properties. The chapter also addresses the method's limitations and presents remedies including line search strategies and the Levenberg-Marquardt modification. We conclude with an introduction to quasi-Newton methods, which approximate the Hessian to reduce computational cost while preserving favorable convergence properties.

Recommended Reading

- Sections 8.6 and 8.7 of Bazaraa, Sherali, and Shetty (2006)
- Sections 2.2, 3.3, and 3.4 of Nocedal and Wright

10.1 Newton's Method: Motivation and Derivation

As before, we consider the **unconstrained optimization problem**:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}). \quad (10.1)$$

Newton's method is based on **minimizing the second-order Taylor**

series approximation of $f(\mathbf{x})$ instead of $f(\mathbf{x})$ directly. Recall that for a twice-differentiable function f , we have the approximation:

$$f(\mathbf{x}) \approx f(\mathbf{x}_k) + \nabla_k^T(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \nabla_k^2(\mathbf{x} - \mathbf{x}_k), \quad (10.2)$$

where we use the notation $\nabla_k := \nabla f(\mathbf{x}_k)$ for the gradient and $\nabla_k^2 := H(\mathbf{x}_k) = \nabla^2 f(\mathbf{x}_k)$ for the Hessian matrix, both evaluated at the current iterate \mathbf{x}_k .

10.1.1 Derivation of the Newton Direction

If the Hessian ∇_k^2 is **positive definite**, then the function $q : \mathbb{R}^n \rightarrow \mathbb{R}$ defined by

$$q(\mathbf{x}) \equiv f(\mathbf{x}_k) + \nabla_k^T(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \nabla_k^2(\mathbf{x} - \mathbf{x}_k) \quad (10.3)$$

is a **convex quadratic function**. Its global minimizer \mathbf{x}^* can be found by setting the gradient of q to zero:

$$\nabla q(\mathbf{x}) = \nabla_k + \nabla_k^2(\mathbf{x} - \mathbf{x}_k) = \mathbf{0}.$$

Solving for \mathbf{x} , we obtain:

$$\mathbf{x}^* = \mathbf{x}_k - (\nabla_k^2)^{-1} \nabla_k. \quad (10.4)$$

Setting $\mathbf{x}_{k+1} = \mathbf{x}^*$, we obtain an iteration of Newton's method:

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla_k^2)^{-1} \nabla_k, \quad k \geq 1.} \quad (10.5)$$

Remark 10.1 (Connection to Other Methods). The Newton iteration (10.5) is a generalization of the one-dimensional Newton's method encountered in line search procedures. It is also closely related to the **Newton-Raphson method** for solving the system of nonlinear equations $\nabla f(\mathbf{x}) = \mathbf{0}$.

10.2 Application to Convex Quadratic Programs

Newton's method exhibits remarkable behavior when applied to convex quadratic functions, demonstrating its power for problems with quadratic structure.

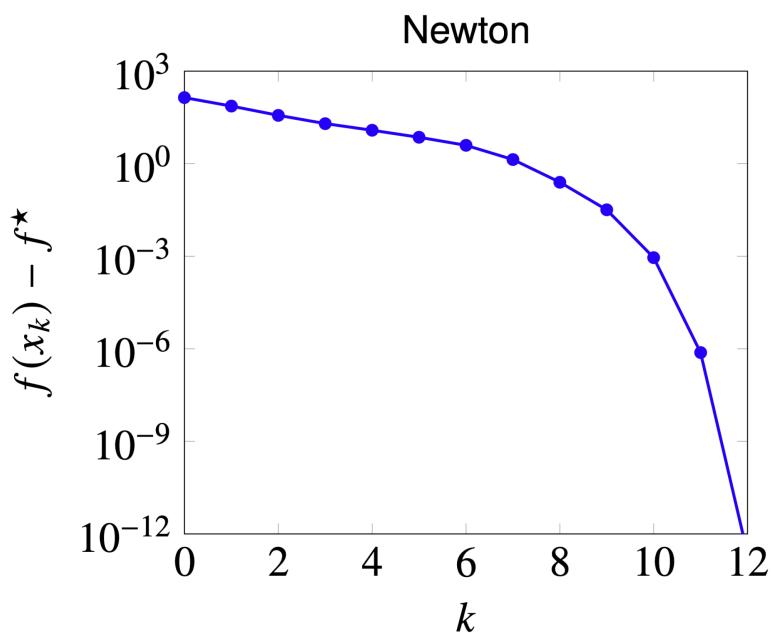


Figure 10.1: Geometric illustration of Newton’s method. At each iteration, the method minimizes a quadratic approximation (based on the second-order Taylor series) to find the next iterate.

Example 10.1 (Newton's Method for Convex Quadratic Functions).

Consider a convex quadratic function of the form:

$$f(\mathbf{x}) \equiv \frac{1}{2} \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}, \quad (10.6)$$

where $Q \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix and $\mathbf{c} \in \mathbb{R}^n$.

For this function:

- The gradient is $\nabla f(\mathbf{x}) = Q\mathbf{x} + \mathbf{c}$
- The Hessian is $\nabla^2 f(\mathbf{x}) = Q$ (constant)

Starting from an initial guess $\mathbf{x}_1 \in \mathbb{R}^n$, the Newton iteration gives:

$$\mathbf{x}_2 = \mathbf{x}_1 - Q^{-1}(Q\mathbf{x}_1 + \mathbf{c}) = \mathbf{x}_1 - \mathbf{x}_1 - Q^{-1}\mathbf{c} = -Q^{-1}\mathbf{c}.$$

We reach the global minimizer in one step starting from *any* initial guess!

This property makes Newton's method particularly attractive: for quadratic functions, it achieves finite termination in a single iteration. For general smooth functions, Newton's method can be viewed as repeatedly solving quadratic approximations.

10.3 Numerical Example

Example 10.2 (Numerical Illustration of Newton's Method). Consider the optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^2} (x_1 - 2)^4 + (x_1 - 2x_2)^2. \quad (10.7)$$

For this function:

- The gradient is:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} 4(x_1 - 2)^3 + 2(x_1 - 2x_2) \\ -4(x_1 - 2x_2) \end{pmatrix}$$

- The Hessian is:

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} 12(x_1 - 2)^2 + 2 & -4 \\ -4 & 8 \end{pmatrix}$$

Starting from $\mathbf{x}_1 = (0, 3)^T$, Newton's method converges to the optimal solution $\mathbf{x}^* = (2, 1)^T$ with $f^* = 0$.

The following table shows the progression of Newton's method:

k	\mathbf{x}_k	$f(\mathbf{x}_k)$	$\ \nabla f(\mathbf{x}_k)\ $
1	(0.000, 3.000)	52.000	40.000
2	(1.333, 0.667)	0.198	0.889
3	(1.778, 0.889)	0.002	0.099
4	(1.963, 0.981)	1.8×10^{-6}	0.006
5	(1.999, 1.000)	1.3×10^{-13}	4.4×10^{-5}

Compare the rate of convergence with the steepest descent method for the same problem (see Table 8.11 of Bazaraa et al.). Newton's method achieves rapid convergence despite the narrow contours of this function that cause steepest descent to zigzag slowly toward the minimum.

Table 8.12 Summary of Computations for the Method of Newton

Iteration k	\mathbf{x}_k $f(\mathbf{x}_k)$	$\nabla f(\mathbf{x}_k)$	$H(\mathbf{x}_k)$	$H(\mathbf{x}_k)^{-1}$	$-H(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$	\mathbf{x}_{k+1}
1	(0.00, 3.00) 52.00	(-44.0, 24.0)	$\begin{bmatrix} 50.0 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{384} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 50.0 \end{bmatrix}$	(0.67, -2.67)	(0.67, 0.33)
2	(0.67, 0.33) 3.13	(-9.39, -0.04)	$\begin{bmatrix} 23.23 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{169.84} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 23.23 \end{bmatrix}$	(0.44, 0.23)	(1.11, 0.56)
3	(1.11, 0.56) 0.63	(-2.84, -0.04)	$\begin{bmatrix} 11.50 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{76} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 11.50 \end{bmatrix}$	(0.30, 0.14)	(1.41, 0.70)
4	(1.41, 0.70) 0.12	(-0.80, -0.04)	$\begin{bmatrix} 6.18 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{33.44} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 6.18 \end{bmatrix}$	(0.20, 0.10)	(1.61, 0.80)
5	(1.61, 0.80) 0.02	(-0.22, -0.04)	$\begin{bmatrix} 3.83 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{14.64} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 3.83 \end{bmatrix}$	(0.13, 0.07)	(1.74, 0.87)
6	(1.74, 0.87) 0.005	(-0.07, 0.00)	$\begin{bmatrix} 2.81 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{6.48} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 2.81 \end{bmatrix}$	(0.09, 0.04)	(1.83, 0.91)
7	(1.83, 0.91) 0.0009	(0.0003, -0.04)				

Figure 10.2: Iteration table showing the progression of Newton's method, demonstrating the rapid convergence characteristic of the algorithm.

10.4 Convergence Properties

Newton's method **deflects the steepest descent direction by premultiplying it by the inverse of the Hessian matrix**. This can be interpreted

Table 8.12 Summary of Computations for the Method of Newton							
Iteration	x_k	$f(x_k)$	$\nabla f(x_k)$	$H(x_k)$	$H(x_k)^{-1}$	$-H(x_k)^{-1}\nabla f(x_k)$	x_{k+1}
1	(0.00, 3.00)	52.00	(-44.0, 24.0)	$\begin{bmatrix} 50.0 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{384} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 50.0 \end{bmatrix}$	(0.67, -2.67)	(0.67, 0.33)
	(0.67, 0.33)	3.13	(-9.34, -0.04)	$\begin{bmatrix} 23.23 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{169.84} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 23.23 \end{bmatrix}$	(0.44, 0.23)	(1.11, 0.56)
3	(1.11, 0.56)	0.63	(-2.84, -0.04)	$\begin{bmatrix} 11.50 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{76} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 11.50 \end{bmatrix}$	(0.30, 0.14)	(1.41, 0.70)
	(1.41, 0.70)	0.12	(-0.80, -0.04)	$\begin{bmatrix} 6.18 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{33.44} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 6.18 \end{bmatrix}$	(0.20, 0.10)	(1.61, 0.80)
5	(1.61, 0.80)	0.02	(-0.22, -0.04)	$\begin{bmatrix} 3.83 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{14.64} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 3.83 \end{bmatrix}$	(0.13, 0.07)	(1.74, 0.87)
	(1.74, 0.87)	0.005	(-0.07, 0.00)	$\begin{bmatrix} 2.81 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{6.48} \begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 2.81 \end{bmatrix}$	(0.09, 0.04)	(1.83, 0.91)
7	(1.83, 0.91)	0.0009	(0.0003, -0.04)				

Figure 10.3: Contour plot illustrating the iterates of Newton's method converging to the optimal solution.

as steepest descent with affine scaling, where the scaling adapts to the local curvature of the objective function.

10.4.1 Potential Issues with Convergence

In general, Newton's method may not converge because of the following issues:

1. **Singular Hessian:** The Hessian matrix ∇_k^2 may be singular, making the Newton direction undefined.
2. **Lack of Descent:** Even if $(\nabla_k^2)^{-1}$ exists, we may have $f(\mathbf{x}_{k+1}) \geq f(\mathbf{x}_k)$, meaning the method fails to reduce the objective value.

10.4.2 Local Quadratic Convergence

Despite these potential issues, Newton's method enjoys excellent local convergence properties under appropriate conditions.

Theorem 10.1 (Local Convergence of Newton's Method). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be twice continuously differentiable. If the initial guess \mathbf{x}_1 is close enough to a stationary point $\bar{\mathbf{x}} \in \mathbb{R}^n$ satisfying:*

- $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$, and
- $\nabla^2 f(\bar{\mathbf{x}})$ is nonsingular (full rank),

then Newton's method is well-defined and converges to $\bar{\mathbf{x}}$ at a quadratic

rate:

$$\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}\| \leq c_1 \|\mathbf{x}_k - \bar{\mathbf{x}}\|^2, \quad (10.8)$$

for some constant $c_1 > 0$.

Remark 10.2. Theorem 8.6.5 of Bazaraa et al. provides the formal statement and proof of this result with explicit conditions on the initial guess.

The quadratic convergence rate (10.8) means that the number of correct digits roughly doubles at each iteration. This is dramatically faster than the linear convergence of steepest descent.

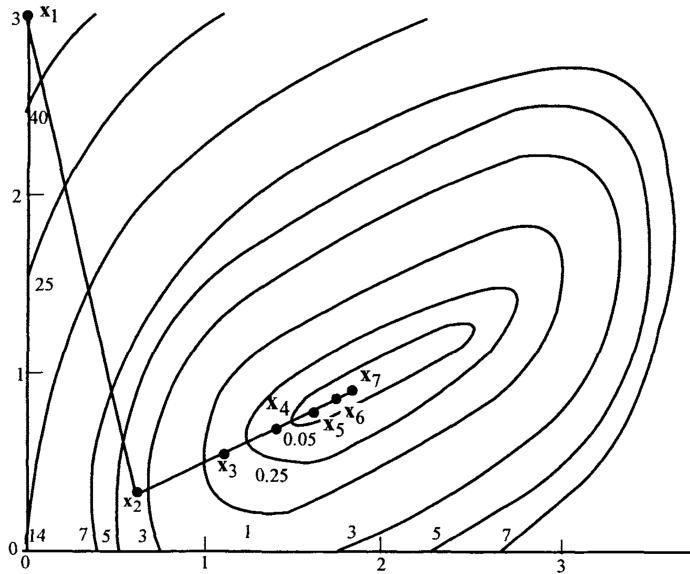


Figure 8.18 Method of Newton.

Figure 10.4: Convergence behavior of Newton's method, illustrating the rapid quadratic convergence near the optimal solution.

10.5 Advantages and Disadvantages

10.5.1 Main Advantage

The primary advantage of Newton's method is its **rapid quadratic convergence** when *all* of the following conditions hold:

- The Hessian matrix $\nabla^2 f(\bar{\mathbf{x}})$ is **positive definite** at the stationary point $\bar{\mathbf{x}} \in \mathbb{R}^n$ to which we wish to converge.
- The initial guess \mathbf{x}_1 is “close enough” to $\bar{\mathbf{x}}$.

10.5.2 Main Disadvantages and Solutions

Newton's method has several disadvantages, each with a corresponding solution:

1. **Lack of the descent property** even with positive definite Hessian matrices ∇_k^2 .

Solution: Use **line search** along the Newton direction.

2. **Indefinite Hessian matrices:** The Hessian matrices ∇_k^2 may not be positive definite, particularly when far from a local minimizer.

Solution: Use the **Levenberg-Marquardt modification**.

3. **Computational cost:** Computing the Hessian ∇_k^2 at each iteration may be expensive, requiring $O(n^2)$ second derivatives.

Solution: Use **quasi-Newton methods** that approximate the Hessian using only gradient information.

10.6 Guaranteeing Descent with Line Search

Consider an iteration of Newton's method where $\nabla_k \neq \mathbf{0}$ and ∇_k^2 is nonsingular:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla_k^2)^{-1} \nabla_k.$$

In general, **Newton's method may not possess the descent property**, i.e., we may have $f(\mathbf{x}_{k+1}) \geq f(\mathbf{x}_k)$.

10.6.1 The Newton Direction as a Descent Direction

However, if ∇_k^2 is positive definite, then the Newton direction

$$\mathbf{d}_k = -(\nabla_k^2)^{-1} \nabla_k \quad (10.9)$$

is a **descent direction** in the sense that there exists $\delta > 0$ such that

$$f(\mathbf{x}_k + \lambda \mathbf{d}_k) < f(\mathbf{x}_k), \quad \forall \lambda \in (0, \delta).$$

Proposition 10.2 (Newton Direction is a Descent Direction). *If $\nabla_k^2 \succ 0$ and $\nabla_k \neq \mathbf{0}$, then $\mathbf{d}_k = -(\nabla_k^2)^{-1} \nabla_k$ is a descent direction.*

Proof. Define the function $\theta_k(\lambda) \equiv f(\mathbf{x}_k + \lambda \mathbf{d}_k)$. Computing its derivative at $\lambda = 0$:

$$\theta'_k(0) = \nabla_k^T \mathbf{d}_k = -\nabla_k^T (\nabla_k^2)^{-1} \nabla_k < 0$$

since $(\nabla_k^2)^{-1}$ is positive definite (the inverse of a positive definite matrix is positive definite) and $\nabla_k \neq \mathbf{0}$.

By the continuity of θ'_k at 0, there exists $\delta > 0$ such that

$$\theta'_k(\lambda) < 0, \quad \forall \lambda \in (0, \delta).$$

Therefore, θ_k is decreasing on $(0, \delta)$, which means

$$f(\mathbf{x}_k + \lambda \mathbf{d}_k) < f(\mathbf{x}_k), \quad \forall \lambda \in (0, \delta).$$

□

10.6.2 Newton's Method with Line Search

Assuming $\nabla_k \neq \mathbf{0}$ and ∇_k^2 is positive definite, we can **perform line search along the Newton direction** $\mathbf{d}_k := -(\nabla_k^2)^{-1} \nabla_k$ to guarantee descent:

- **Line search:** Pick $\lambda_k \in \arg \min_{\lambda \geq 0} f\left(\mathbf{x}_k - \lambda (\nabla_k^2)^{-1} \nabla_k\right)$
- **Update:** Set $\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k (\nabla_k^2)^{-1} \nabla_k$

The descent property follows from the observation that

$$f\left(\mathbf{x}_k - \lambda (\nabla_k^2)^{-1} \nabla_k\right) < f(\mathbf{x}_k), \quad \forall \lambda \in (0, \delta).$$

Remark 10.3 (Backtracking Line Search). In practice, **backtracking line search** is often used: start with step length $\bar{\lambda} = 1$ (the pure Newton step); reduce the step length if no sufficient decrease is achieved. This combines the fast local convergence of Newton's method (when $\lambda_k = 1$ is accepted) with guaranteed global convergence.

10.7 Levenberg-Marquardt Modification

Newton's method assumes that the Hessian ∇_k^2 is positive definite. When this assumption fails, we need a modification to ensure the method remains well-defined and produces descent directions.

10.7.1 The Modified Hessian

If the Hessian ∇_k^2 is not positive definite, it can be modified so that the Newton iteration resulting from this modification possesses the descent property. Consider the matrix:

$$M_k := \nabla_k^2 + \mu_k I_n, \quad (10.10)$$

where I_n is the $n \times n$ identity matrix and $\mu_k \geq 0$ is a regularization parameter.

Proposition 10.3 (Positive Definiteness of Modified Hessian). *If we choose μ_k to be a sufficiently large positive value, then M_k will be a positive definite matrix. Specifically, picking*

$$\mu_k > |\alpha_{\min}(\nabla_k^2)|,$$

where $\alpha_{\min}(\nabla_k^2)$ denotes the minimum eigenvalue of ∇_k^2 , ensures that M_k is positive definite.

Proof. Let $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ be the eigenvalues of ∇_k^2 . The eigenvalues of $M_k = \nabla_k^2 + \mu_k I_n$ are $\lambda_i + \mu_k$ for $i = 1, \dots, n$. If $\mu_k > |\lambda_1| = |\alpha_{\min}(\nabla_k^2)|$, then all eigenvalues of M_k are positive, making M_k positive definite. \square

10.7.2 The Levenberg-Marquardt Update

To ensure descent, we use the direction $-M_k^{-1}\nabla_k$ within Newton's method instead of $-(\nabla_k^2)^{-1}\nabla_k$.

By performing (exact) line search along the direction $-M_k^{-1}\nabla_k$, we obtain the following update:

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k M_k^{-1} \nabla_k,} \quad (10.11)$$

where $\lambda_k \in \arg \min_{\lambda \geq 0} f(\mathbf{x}_k - \lambda M_k^{-1} \nabla_k)$.

This is referred to as the **Levenberg-Marquardt modification**.

Proposition 10.4 (Descent Property of Levenberg-Marquardt Direction). *The direction $-M_k^{-1}\nabla_k$ is a descent direction when M_k is positive definite and $\nabla_k \neq \mathbf{0}$.*

Proof. Define the function $\theta_k(\lambda) \equiv f(\mathbf{x}_k - \lambda M_k^{-1} \nabla_k)$. Since M_k^{-1} is positive definite (as the inverse of a positive definite matrix) and $\nabla_k \neq \mathbf{0}$, we have

$$\theta'_k(0) = -\nabla_k^T (M_k)^{-1} \nabla_k < 0.$$

The result follows by continuity, as in the proof of Proposition 10.2. \square

10.7.3 Interpolation Between Methods

The Levenberg-Marquardt modification is, in some sense, **between steepest descent and Newton's method**:

- If ∇_k^2 is positive definite, we can set $\mu_k = 0$ and the Levenberg-Marquardt method **coincides with Newton's method**.
- On the other hand, if we set μ_k to a large positive value, then $M_k \approx \mu_k I_n$, so $M_k^{-1} \approx \mu_k^{-1} I_n$ and the update becomes:

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k - \lambda_k \mu_k^{-1} \nabla_k.$$

Thus, when μ_k is set to a large positive value, we obtain an **approximation of the steepest descent iteration**.

Remark 10.4 (Global Convergence). Under suitable assumptions, Newton's method with the Levenberg-Marquardt modification enjoys *global* convergence, meaning it converges to a stationary point from any starting point.

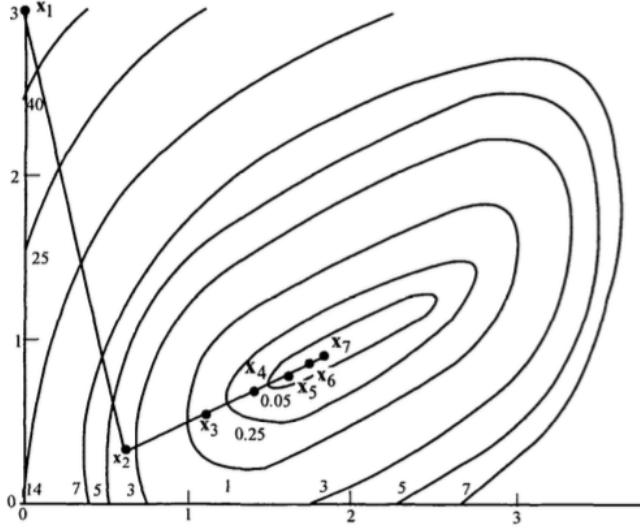


Figure 10.5: Illustration of the Levenberg-Marquardt modification showing how the method interpolates between Newton's method and steepest descent depending on the regularization parameter μ_k .

10.8 Quasi-Newton Methods

10.8.1 Motivation

The iteration of Newton's method (assuming $\nabla_k \neq \mathbf{0}$ and $\nabla_k^2 \succ 0$) is:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla_k^2)^{-1} \nabla_k.$$

The **major computational steps in Newton's method** are:

1. Computing the Hessian ∇_k^2 — requires $O(n^2)$ second derivatives.
2. Solving the linear system $\nabla_k^2(\mathbf{x}_{k+1} - \mathbf{x}_k) = -\nabla_k$ — requires $O(n^3)$ operations.

Definition 10.1 (Quasi-Newton Methods). **Quasi-Newton methods** approximate the Hessian ∇_k^2 using a matrix B_k *without* computing second derivatives:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - B_k^{-1} \nabla_k. \quad (10.12)$$

The approximation B_k is updated after each step using only gradient

information.

10.8.2 The Secant Condition

How do we approximate the Hessian ∇_k^2 using only first derivatives?

From the mean value theorem, for smooth functions we have the approximation:

$$\nabla_k^2(\mathbf{x}_{k+1} - \mathbf{x}_k) \approx (\nabla_{k+1} - \nabla_k).$$

We can require the *new* Hessian approximation B_{k+1} to satisfy the **secant condition**:

$$B_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k) = (\nabla_{k+1} - \nabla_k). \quad (10.13)$$

Additional desirable properties for B_{k+1} :

- B_{k+1} should be a **symmetric** matrix.
- B_{k+1} should be “close to” B_k (minimal change).
- $B_k \succ 0 \implies B_{k+1} \succ 0$ (preservation of positive definiteness).

Ultimately, we want an approximation B_{k+1} such that:

- We can easily update B_{k+1} from B_k .
- We can easily solve the linear system $B_{k+1}(\mathbf{x}_{k+2} - \mathbf{x}_{k+1}) = -\nabla_{k+1}$.

10.8.3 Popular Quasi-Newton Update Formulae

Define the following notation for convenience:

$$\mathbf{s}_k := \mathbf{x}_{k+1} - \mathbf{x}_k, \quad \mathbf{y}_k := \nabla_{k+1} - \nabla_k.$$

Symmetric-Rank-One (SR1) Formula

$$B_{k+1} := B_k + \frac{(\mathbf{y}_k - B_k \mathbf{s}_k)(\mathbf{y}_k - B_k \mathbf{s}_k)^T}{(\mathbf{y}_k - B_k \mathbf{s}_k)^T \mathbf{s}_k}. \quad (10.14)$$

Disadvantage: $B_k \succ 0$ does *not* imply $B_{k+1} \succ 0$. The SR1 update may produce indefinite matrices.

BFGS Formula

The BFGS formula, named after **B**royden, **F**letcher, **G**oldfarb, and **S**hanno, is:

$$B_{k+1} := B_k - \frac{B_k \mathbf{s}_k \mathbf{s}_k^T B_k}{\mathbf{s}_k^T B_k \mathbf{s}_k} + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}. \quad (10.15)$$

Advantage: Under the condition $\mathbf{y}_k^T \mathbf{s}_k > 0$ (which is satisfied when using line search with the Wolfe conditions), we have:

$$B_k \succ 0 \implies B_{k+1} \succ 0.$$

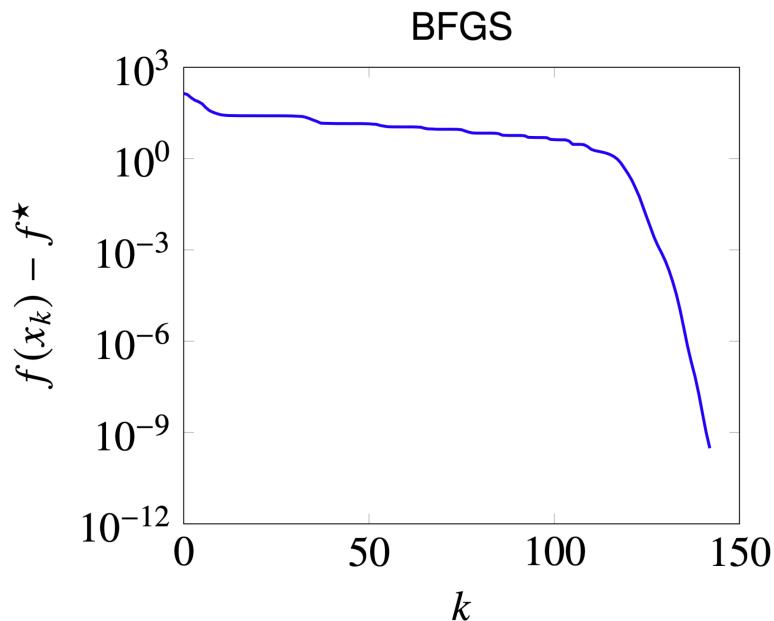


Figure 10.6: Illustration of the BFGS update formula, showing how the Hessian approximation B_k is updated using gradient differences to form B_{k+1} .

Remark 10.5 (Practical Implementation). In practice, quasi-Newton methods update $(B_{k+1})^{-1}$ directly from $(B_k)^{-1}$ using the Sherman-Morrison-Woodbury formula. This allows direct application of the search direction $\mathbf{d}_{k+1} = -B_{k+1}^{-1} \nabla_{k+1}$ without solving a linear system. See Section 2.2 or Chapter 6 of Nocedal and Wright for details.

10.8.4 Computational Comparison: Newton vs. BFGS

Example 10.3 (Newton versus BFGS). Consider the following optimization problem with $n = 100$ variables and $m = 500$ constraints:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{c}^T \mathbf{x} - \sum_{i=1}^m \ln(b_i - \mathbf{a}_i^T \mathbf{x}).$$

The computational costs per iteration are:

- **Newton's method:** $O(n^3)$ per iteration (for solving the linear system with the Hessian).
- **BFGS:** $O(n^2)$ per iteration (for the matrix-vector products in the update).

While Newton's method typically requires fewer iterations due to its quadratic convergence, the lower per-iteration cost of BFGS often makes it more efficient for large-scale problems.

Remark 10.6 (When to Use Each Method). • Use **Newton's method** when:

- The Hessian is cheap to compute (e.g., sparse structure).
 - The problem dimension is moderate.
 - Very high accuracy is required.
- Use **quasi-Newton methods (BFGS)** when:
- Computing the Hessian is expensive or infeasible.
 - The problem dimension is large.
 - Moderate accuracy is sufficient.

10.9 Summary

This chapter covered the following key concepts:

1. **Newton's Method** minimizes a second-order Taylor approximation of the objective function, leading to the iteration $\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla_k^2)^{-1} \nabla_k$.

2. For **convex quadratic functions**, Newton's method converges in one step from any starting point.
3. Newton's method achieves **quadratic convergence** locally near a stationary point with a nonsingular Hessian.
4. **Line search** can be combined with Newton's method to guarantee descent when the Hessian is positive definite.
5. The **Levenberg-Marquardt modification** handles indefinite Hessians by adding a multiple of the identity matrix, interpolating between Newton's method and steepest descent.
6. **Quasi-Newton methods** (such as BFGS) approximate the Hessian using only gradient information, reducing the per-iteration cost from $O(n^3)$ to $O(n^2)$ while maintaining superlinear convergence.

Chapter 11

Conjugate Direction and Conjugate Gradient Methods

This chapter develops conjugate direction methods for solving optimization problems, particularly convex quadratic programs. We begin with the theory of conjugate directions and their key properties, then present the conjugate direction method and establish its finite convergence for quadratic functions. We conclude with the celebrated conjugate gradient method and its extensions to nonlinear problems.

Recommended Reading

- Section 8.8 of Bazaraa, Sherali, and Shetty (2006)
- Chapter 5 of Nocedal and Wright
- **Supplementary:** Section 4.4 of Wright and Recht (2022) — connections between conjugate gradient and momentum methods

11.1 Conjugate Directions

The concept of conjugate directions provides a powerful framework for solving quadratic optimization problems. Unlike steepest descent, which can zigzag toward the solution, conjugate direction methods ensure that progress made in each direction is never undone by subsequent iterations.

11.1.1 Definition and Basic Properties

Definition 11.1 (*Q*-Conjugate Directions). Given an $n \times n$ *positive definite* matrix Q , the nonzero directions $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k \in \mathbb{R}^n$ are said to be ***Q*-conjugate** (or **conjugate with respect to *Q***) if

$$\mathbf{d}_i^T Q \mathbf{d}_j = 0, \quad \forall i, j \in \{1, \dots, k\} \text{ with } i \neq j.$$

When $Q = I_n$ (the identity matrix), *Q*-conjugacy reduces to ordinary orthogonality. Thus, conjugacy can be viewed as a generalization of orthogonality that accounts for the curvature of the quadratic function defined by Q .

Theorem 11.1 (Linear Independence of Conjugate Directions). *Any set of *Q*-conjugate directions $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k$ is linearly independent.*

Proof. Let $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k$ be *Q*-conjugate directions. To prove linear independence, we show that the only solution of the system

$$\alpha_1 \mathbf{d}_1 + \alpha_2 \mathbf{d}_2 + \cdots + \alpha_k \mathbf{d}_k = \mathbf{0}$$

in the scalars $\alpha_1, \alpha_2, \dots, \alpha_k$ is $\alpha_1 = \alpha_2 = \cdots = \alpha_k = 0$.

Multiplying both sides of the equation by $\mathbf{d}_i^T Q$ from the left yields

$$\alpha_1 \mathbf{d}_i^T Q \mathbf{d}_1 + \alpha_2 \mathbf{d}_i^T Q \mathbf{d}_2 + \cdots + \alpha_i \mathbf{d}_i^T Q \mathbf{d}_i + \cdots + \alpha_k \mathbf{d}_i^T Q \mathbf{d}_k = 0.$$

Since $\mathbf{d}_1, \dots, \mathbf{d}_k$ are *Q*-conjugate, all terms with $j \neq i$ vanish, leaving

$$\alpha_i \mathbf{d}_i^T Q \mathbf{d}_i = 0.$$

Since Q is positive definite and $\mathbf{d}_i \neq \mathbf{0}$, we have $\mathbf{d}_i^T Q \mathbf{d}_i > 0$. Therefore, $\alpha_i = 0$. Since $i \in \{1, \dots, k\}$ was chosen arbitrarily, we conclude that $\alpha_1 = \cdots = \alpha_k = 0$. \square

Corollary 11.2. *We can choose at most n *Q*-conjugate directions in \mathbb{R}^n . Moreover, given n *Q*-conjugate directions $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n$, we can express any $\mathbf{x} \in \mathbb{R}^n$ as*

$$\mathbf{x} = \sum_{i=1}^n \beta_i \mathbf{d}_i$$

for some scalars $\beta_1, \beta_2, \dots, \beta_n \in \mathbb{R}$.

Remark 11.1 (Generating Conjugate Directions). Several methods exist for generating conjugate directions:

1. **Eigendecomposition of Q :** The eigenvectors of a symmetric positive definite matrix Q are mutually orthogonal and hence Q -conjugate.
2. **Conjugate Gram-Schmidt process:** Given linearly independent vectors $\mathbf{u}_1, \dots, \mathbf{u}_n$, one can construct Q -conjugate directions analogously to the classical Gram-Schmidt orthogonalization.
3. **Conjugate gradient method:** An elegant approach that generates conjugate directions iteratively using only gradient information (discussed in Section 11.6).

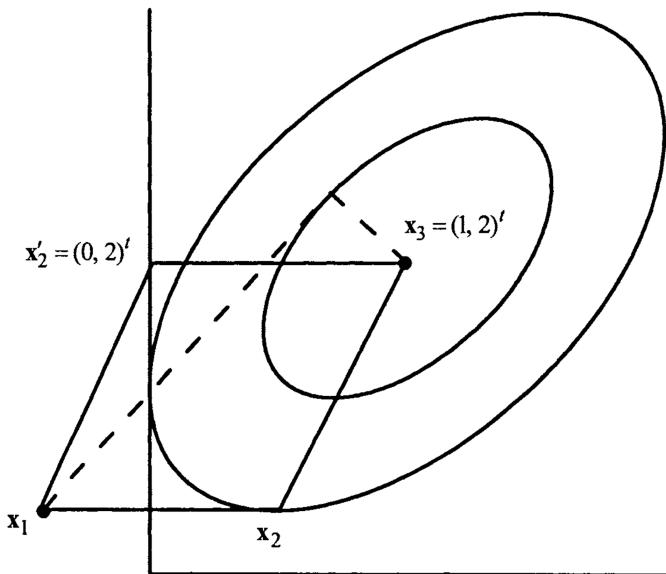


Figure 8.20 Illustration of conjugate directions.

Figure 11.1: Illustration of conjugate directions and the conjugate gradient method. The ellipses represent level curves of a quadratic function $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}$. Conjugate directions allow the method to reach the minimizer without the zigzagging behavior characteristic of steepest descent on ill-conditioned problems.

11.2 Conjugate Direction Method for Convex Quadratic Programs

We now present the conjugate direction method for minimizing convex quadratic functions.

11.2.1 Problem Setup

Consider the **convex quadratic program**

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}, \quad (11.1)$$

where Q is an $n \times n$ positive definite matrix and $\mathbf{c} \in \mathbb{R}^n$.

Since Q is positive definite, f is strictly convex, and the unique global minimizer is given by the solution of $\nabla f(\mathbf{x}^*) = Q\mathbf{x}^* + \mathbf{c} = \mathbf{0}$, i.e., $\mathbf{x}^* = -Q^{-1}\mathbf{c}$.

11.2.2 Algorithm Description

Definition 11.2 (Conjugate Direction Method). Given a starting point $\mathbf{x}_1 \in \mathbb{R}^n$ and n Q -conjugate directions $\mathbf{d}_1, \dots, \mathbf{d}_n \in \mathbb{R}^n$, the **conjugate direction method** generates a sequence of iterates $\{\mathbf{x}_k\}_{k=1}^{n+1}$ defined by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k, \quad \forall k \in \{1, \dots, n\},$$

where λ_k is the solution of the exact line search subproblem

$$\lambda_k \in \operatorname{argmin}_{\lambda \in \mathbb{R}} f(\mathbf{x}_k + \lambda \mathbf{d}_k).$$

11.2.3 Derivation of the Step Length

Let $\theta_k(\lambda) := f(\mathbf{x}_k + \lambda \mathbf{d}_k)$ denote the line search objective. Expanding this expression:

$$\begin{aligned} \theta_k(\lambda) &= \frac{1}{2} (\mathbf{x}_k + \lambda \mathbf{d}_k)^T Q (\mathbf{x}_k + \lambda \mathbf{d}_k) + \mathbf{c}^T (\mathbf{x}_k + \lambda \mathbf{d}_k) \\ &= \frac{1}{2} \lambda^2 \mathbf{d}_k^T Q \mathbf{d}_k + \lambda (\mathbf{x}_k^T Q + \mathbf{c}^T) \mathbf{d}_k + f(\mathbf{x}_k) \\ &= \frac{1}{2} \lambda^2 \mathbf{d}_k^T Q \mathbf{d}_k + \lambda \nabla f(\mathbf{x}_k)^T \mathbf{d}_k + f(\mathbf{x}_k), \end{aligned}$$

where $\nabla f(\mathbf{x}_k) = Q\mathbf{x}_k + \mathbf{c}$ denotes the gradient at \mathbf{x}_k .

Since θ_k is a convex quadratic in λ (with positive coefficient $\frac{1}{2}\mathbf{d}_k^T Q \mathbf{d}_k > 0$), the optimal step length is found by setting $\theta'_k(\lambda_k) = 0$:

$$\theta'_k(\lambda_k) = \lambda_k \mathbf{d}_k^T Q \mathbf{d}_k + \nabla f(\mathbf{x}_k)^T \mathbf{d}_k = 0.$$

Solving for λ_k :

$$\lambda_k = -\frac{\nabla f(\mathbf{x}_k)^T \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k}. \quad (11.2)$$

Hence, iteration k of the conjugate direction method reduces to:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\nabla f(\mathbf{x}_k)^T \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k} \mathbf{d}_k. \quad (11.3)$$

Remark 11.2 (Comparison with Other Methods).

• **Steepest descent:**

Uses $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$ as the search direction. For ill-conditioned problems (large condition number of Q), steepest descent can exhibit slow convergence due to zigzagging.

- **Newton's method:** Uses $\mathbf{d}_k = -Q^{-1}\nabla f(\mathbf{x}_k)$, which gives $\mathbf{x}_{k+1} = \mathbf{x}^*$ in one step for quadratic functions. However, this requires computing and inverting the Hessian.
- **Conjugate directions:** Achieves exact convergence in n steps without requiring Hessian inversion, offering a compromise between the simplicity of steepest descent and the rapid convergence of Newton's method.

11.3 Properties of the Conjugate Direction Method

11.3.1 Orthogonality of Gradients to Previous Directions

Theorem 11.3 (Gradient Orthogonality). *Let $\nabla_k := \nabla f(\mathbf{x}_k)$ denote the gradient at iterate \mathbf{x}_k . Then*

$$\nabla_{k+1}^T \mathbf{d}_i = 0, \quad \forall i \in \{1, \dots, k\}.$$

That is, the gradient at \mathbf{x}_{k+1} is orthogonal to all previous search directions.

Proof. We prove this by induction on k .

Base case: We first show that $\nabla_{k+1}^T \mathbf{d}_k = 0$ for each $k \in \{1, \dots, n\}$. Since λ_k minimizes $\theta_k(\lambda) = f(\mathbf{x}_k + \lambda \mathbf{d}_k)$, the first-order necessary optimality condition gives

$$\theta'_k(\lambda_k) = 0.$$

Using the chain rule:

$$\theta'_k(\lambda_k) = \frac{d}{d\lambda} f(\mathbf{x}_k + \lambda \mathbf{d}_k) \Big|_{\lambda=\lambda_k} = \nabla f(\mathbf{x}_k + \lambda_k \mathbf{d}_k)^T \mathbf{d}_k = \nabla_{k+1}^T \mathbf{d}_k = 0.$$

This establishes the base case.

Induction hypothesis: Suppose the result holds for $k - 1$, i.e.,

$$\nabla_k^T \mathbf{d}_i = 0, \quad \forall i \in \{1, \dots, k - 1\}.$$

Induction step: We wish to show that $\nabla_{k+1}^T \mathbf{d}_i = 0$ for all $i \in \{1, \dots, k\}$. The case $i = k$ was already established above. For $i \in \{1, \dots, k - 1\}$, we compute:

$$\begin{aligned} \nabla_{k+1}^T \mathbf{d}_i &= (Q\mathbf{x}_{k+1} + \mathbf{c})^T \mathbf{d}_i \\ &= (Q(\mathbf{x}_k + \lambda_k \mathbf{d}_k) + \mathbf{c})^T \mathbf{d}_i \\ &= (Q\mathbf{x}_k + \mathbf{c} + \lambda_k Q\mathbf{d}_k)^T \mathbf{d}_i \\ &= (\nabla_k + \lambda_k Q\mathbf{d}_k)^T \mathbf{d}_i \\ &= \nabla_k^T \mathbf{d}_i + \lambda_k \mathbf{d}_k^T Q\mathbf{d}_i \\ &= 0 + 0 = 0, \end{aligned}$$

where the first term vanishes by the induction hypothesis and the second term vanishes by Q -conjugacy of the directions. \square

11.3.2 Invariance of Step Lengths

Theorem 11.4 (Step Length Invariance). *Irrespective of the order in which we use the Q -conjugate directions $\mathbf{d}_1, \dots, \mathbf{d}_n$, the conjugate direction method takes the same step lengths $\lambda_1, \dots, \lambda_n$ in these directions (though possibly in a different order).*

This theorem implies that the conjugate direction method converges in n iterations, as we now establish.

11.3.3 Expanding Subspace Minimization

Let $D^k = [\mathbf{d}_1 \ \mathbf{d}_2 \ \cdots \ \mathbf{d}_k]$ be the $n \times k$ matrix whose columns are the first k search directions, and let $\boldsymbol{\alpha}^k := (\alpha_1, \alpha_2, \dots, \alpha_k)^T \in \mathbb{R}^k$. Define

$$\theta_k(\boldsymbol{\alpha}^k) := f(\mathbf{x}_1 + D^k \boldsymbol{\alpha}^k) = f\left(\mathbf{x}_1 + \sum_{i=1}^k \alpha_i \mathbf{d}_i\right).$$

Theorem 11.5 (Expanding Subspace Property). *Let $\boldsymbol{\lambda}^k := (\lambda_1, \lambda_2, \dots, \lambda_k)^T$ be the vector of step lengths taken by the conjugate direction method. Then*

$$\boldsymbol{\lambda}^k \in \operatorname{argmin}_{\boldsymbol{\alpha}^k \in \mathbb{R}^k} \theta_k(\boldsymbol{\alpha}^k).$$

That is, after k iterations, the conjugate direction method has minimized f over the affine subspace $\mathbf{x}_1 + \operatorname{span}(\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k)$.

Proof. Expanding the objective function:

$$\begin{aligned}\theta_k(\boldsymbol{\alpha}^k) &= \frac{1}{2}(\mathbf{x}_1 + D^k \boldsymbol{\alpha}^k)^T Q(\mathbf{x}_1 + D^k \boldsymbol{\alpha}^k) + \mathbf{c}^T(\mathbf{x}_1 + D^k \boldsymbol{\alpha}^k) \\ &= \frac{1}{2}(\boldsymbol{\alpha}^k)^T ((D^k)^T Q D^k) \boldsymbol{\alpha}^k + (\mathbf{x}_1^T Q D^k + \mathbf{c}^T D^k) \boldsymbol{\alpha}^k + f(\mathbf{x}_1).\end{aligned}$$

Since Q is positive definite and D^k has full column rank (because the Q -conjugate directions are linearly independent), the matrix $(D^k)^T Q D^k$ is positive definite. Therefore, $\theta_k(\boldsymbol{\alpha}^k)$ is a strictly convex quadratic function with a unique minimizer.

Computing the gradient of θ_k at $\boldsymbol{\alpha}^k = \boldsymbol{\lambda}^k$:

$$\begin{aligned}\nabla \theta_k(\boldsymbol{\lambda}^k)^T &= \nabla f(\mathbf{x}_1 + D^k \boldsymbol{\lambda}^k)^T D^k \\ &= \nabla_{k+1}^T D^k \\ &= \nabla_{k+1}^T [\mathbf{d}_1 \ \mathbf{d}_2 \ \cdots \ \mathbf{d}_k] \\ &= [\nabla_{k+1}^T \mathbf{d}_1 \ \nabla_{k+1}^T \mathbf{d}_2 \ \cdots \ \nabla_{k+1}^T \mathbf{d}_k] \\ &= \mathbf{0}^T,\end{aligned}$$

where the last equality follows from Theorem 11.3.

Since θ_k is strictly convex and $\nabla \theta_k(\boldsymbol{\lambda}^k) = \mathbf{0}$, we conclude that $\boldsymbol{\lambda}^k$ is the unique global minimizer of θ_k . \square

11.4 Convergence of the Conjugate Direction Method

Theorem 11.6 (Finite Convergence). *For a convex quadratic function with a positive definite Hessian, the conjugate direction method converges to the global minimizer in no more than n iterations starting from any point $\mathbf{x}_1 \in \mathbb{R}^n$.*

Proof. Since the n Q -conjugate directions $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n$ are linearly independent (by Theorem 11.1), they form a basis for \mathbb{R}^n . Therefore, any $\mathbf{x} \in \mathbb{R}^n$ can be written as

$$\mathbf{x} = \mathbf{x}_1 + \sum_{i=1}^n \alpha_i \mathbf{d}_i = \mathbf{x}_1 + D^n \boldsymbol{\alpha}^n$$

for some $\boldsymbol{\alpha}^n \in \mathbb{R}^n$. This means

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) = \min_{\boldsymbol{\alpha}^n \in \mathbb{R}^n} f(\mathbf{x}_1 + D^n \boldsymbol{\alpha}^n) = \min_{\boldsymbol{\alpha}^n \in \mathbb{R}^n} \theta_n(\boldsymbol{\alpha}^n).$$

By Theorem 11.5, $\boldsymbol{\lambda}^n$ minimizes $\theta_n(\boldsymbol{\alpha}^n)$ over \mathbb{R}^n . Therefore,

$$f(\mathbf{x}_{n+1}) = f(\mathbf{x}_1 + D^n \boldsymbol{\lambda}^n) = \theta_n(\boldsymbol{\lambda}^n) = \min_{\boldsymbol{\alpha}^n \in \mathbb{R}^n} \theta_n(\boldsymbol{\alpha}^n) = \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}).$$

Hence, \mathbf{x}_{n+1} is the global minimizer of f . □

Remark 11.3 (Expanding Subspace Theorem). The results of the previous section show that after $1 \leq k \leq n$ iterations of the conjugate direction method, we have minimized $f(\mathbf{x})$ on the translated subspace

$$\{\mathbf{x}_1\} + \text{span}(\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k).$$

This property is often called the **expanding subspace theorem** or **expanding manifold theorem**. At each iteration, the method extends minimization to an additional dimension, ensuring that no progress is lost.

11.5 Numerical Example

Example 11.1 (Conjugate Direction Method). Consider the unconstrained quadratic minimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^2} f(\mathbf{x}) = 4x_1^2 + 4x_2^2 - 4x_1x_2 - 12x_2.$$

We can write this in standard form $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}$ with

$$Q = \begin{pmatrix} 8 & -4 \\ -4 & 8 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 0 \\ -12 \end{pmatrix}.$$

The matrix Q is positive definite (its eigenvalues are 4 and 12), so the function is strictly convex with a unique global minimizer.

Optimal solution: Setting $\nabla f(\mathbf{x}^*) = Q\mathbf{x}^* + \mathbf{c} = \mathbf{0}$ gives $\mathbf{x}^* = -Q^{-1}\mathbf{c} = (1, 2)^T$.

Conjugate direction method: Starting from $\mathbf{x}_1 = (0, 0)^T$ with Q -conjugate directions $\mathbf{d}_1 = (1, 0)^T$ and $\mathbf{d}_2 = (1, 2)^T$ (which satisfy $\mathbf{d}_1^T Q \mathbf{d}_2 = 0$), the method converges to $\mathbf{x}^* = (1, 2)^T$ in exactly 2 iterations.

This example illustrates how the conjugate direction method solves the 2-dimensional problem in exactly 2 iterations, as guaranteed by Theorem 11.6. Compare this with steepest descent, which may require many more iterations for ill-conditioned problems.

11.6 The Conjugate Gradient Method

The conjugate direction method requires specifying n Q -conjugate directions in advance. The **conjugate gradient (CG) method** is a special conjugate direction method that generates the conjugate directions iteratively using only gradient information. Its key advantage is that computing a new direction \mathbf{d}_k requires only the previous direction \mathbf{d}_{k-1} , rather than all previous directions.

11.6.1 Algorithm Description

Definition 11.3 (Conjugate Gradient Method). Given a starting point $\mathbf{x}_1 \in \mathbb{R}^n$, the conjugate gradient method generates iterates as follows:

1. **Initialize:** Set $\mathbf{d}_1 = -\nabla f(\mathbf{x}_1) = -\nabla_1$.

2. **For** $k = 1, 2, \dots, n$:

(a) Compute the step length:

$$\lambda_k = -\frac{\nabla_k^T \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k}$$

(b) Update the iterate:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$$

(c) Compute the coefficient:

$$\beta_k = \frac{\nabla_{k+1}^T Q \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k}$$

(d) Update the direction:

$$\mathbf{d}_{k+1} = -\nabla_{k+1} + \beta_k \mathbf{d}_k$$

The method can be terminated early when $\|\nabla f(\mathbf{x}_k)\| < \epsilon$ for some tolerance $\epsilon > 0$.

11.6.2 Derivation of the Direction Update

The key insight is to choose each new direction \mathbf{d}_{k+1} as a linear combination of the negative gradient $-\nabla_{k+1}$ and the previous direction \mathbf{d}_k :

$$\mathbf{d}_{k+1} = -\nabla_{k+1} + \beta_k \mathbf{d}_k.$$

We select β_k to ensure Q -conjugacy: $\mathbf{d}_{k+1}^T Q \mathbf{d}_k = 0$. Substituting and

solving:

$$\begin{aligned} (-\nabla_{k+1} + \beta_k \mathbf{d}_k)^T Q \mathbf{d}_k &= 0 \\ -\nabla_{k+1}^T Q \mathbf{d}_k + \beta_k \mathbf{d}_k^T Q \mathbf{d}_k &= 0 \\ \beta_k &= \frac{\nabla_{k+1}^T Q \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k}. \end{aligned}$$

Theorem 11.7 (Conjugacy of CG Directions). *The directions $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n$ generated by the conjugate gradient method are Q -conjugate. Therefore, the conjugate gradient method converges in at most n iterations.*

Proof. See Theorem 5.3 of Nocedal and Wright for the detailed proof by induction. \square

Remark 11.4 (Computational Advantages). The conjugate gradient method has several attractive properties:

- Each iteration requires only one matrix-vector product $Q \mathbf{d}_k$.
- Storage is minimal: only the current iterate, gradient, and search direction need to be maintained.
- For sparse matrices Q , the method is particularly efficient.
- The method is often effective even when terminated early (before n iterations).

See Algorithms 5.1 and 5.2 of Nocedal and Wright for implementation details.

11.7 Nonlinear Conjugate Gradient Methods

The conjugate gradient method can be extended to minimize general (non-quadratic) smooth functions. The basic idea is to apply the CG framework using local quadratic approximations.

11.7.1 Motivation

If the function f to be minimized is not quadratic, one can use its second-order Taylor series approximation around the current iterate \mathbf{x}_k :

$$f(\mathbf{x}) \approx f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_k)^T \nabla^2 f(\mathbf{x}_k) (\mathbf{x} - \mathbf{x}_k).$$

This quadratic approximation can be minimized using the conjugate gradient method to obtain the next iterate \mathbf{x}_{k+1} .

11.7.2 Practical Modifications

In the standard CG method for quadratics, computing λ_k and β_k requires knowing the Hessian matrix Q . For general nonlinear functions, this is computationally expensive. Practical nonlinear CG methods make the following modifications:

1. **Line search for λ_k :** Instead of using the closed-form expression (11.2), approximate λ_k using a line search procedure (e.g., backtracking or Wolfe conditions).
2. **Hessian-free formulas for β_k :** Several formulas have been proposed that depend only on gradients:

- **Fletcher-Reeves:**

$$\beta_k^{FR} = \frac{\nabla_{k+1}^T \nabla_{k+1}}{\nabla_k^T \nabla_k}$$

- **Polak-Ribière:**

$$\beta_k^{PR} = \frac{\nabla_{k+1}^T (\nabla_{k+1} - \nabla_k)}{\nabla_k^T \nabla_k}$$

- **Polak-Ribière⁺:** $\beta_k^{PR+} = \max\{0, \beta_k^{PR}\}$

Remark 11.5. For strictly quadratic functions, all these formulas reduce to the standard CG formula. For general nonlinear functions, the Polak-Ribière formula often performs better in practice, as it has a built-in restart mechanism when $\nabla_{k+1} \approx \nabla_k$.

11.7.3 Nonlinear CG Algorithm

The general nonlinear conjugate gradient algorithm proceeds as follows:

1. **Initialize:** Set $\mathbf{d}_1 = -\nabla f(\mathbf{x}_1)$.
2. **For** $k = 1, 2, \dots$:
 - (a) Compute λ_k via line search satisfying appropriate conditions (e.g., Wolfe conditions).
 - (b) Update: $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$.
 - (c) Compute β_k using Fletcher-Reeves, Polak-Ribière, or another formula.
 - (d) Update direction: $\mathbf{d}_{k+1} = -\nabla_{k+1} + \beta_k \mathbf{d}_k$.
 - (e) If $\|\nabla_{k+1}\| < \epsilon$, terminate.

For nonlinear problems, it is common to restart the algorithm (by setting $\mathbf{d}_k = -\nabla_k$) periodically or when the directions appear to lose conjugacy. See Algorithm 5.4 of Nocedal and Wright for a complete description and convergence analysis.

Remark 11.6 (Convergence Properties). For general smooth functions:

- The nonlinear CG method with Fletcher-Reeves and strong Wolfe line search is globally convergent.
- The Polak-Ribière⁺ variant is also globally convergent with appropriate line search conditions.
- Near a local minimizer where the Hessian is well-conditioned, nonlinear CG methods exhibit superlinear convergence.

Chapter 12

Modern First-Order Methods

This chapter introduces modern first-order optimization methods that have become essential tools in machine learning and data science. We begin with **Stochastic Gradient Descent** (SGD), the workhorse algorithm for training large-scale machine learning models. We then present **momentum methods**, including Nesterov’s accelerated gradient, which achieve provably faster convergence rates. Finally, we introduce **proximal gradient methods** for handling nonsmooth regularizers such as the ℓ_1 norm used in LASSO and sparse optimization. These methods build on the foundations developed in Chapters 9–11 and connect classical optimization theory to contemporary practice.

Recommended Reading

- **Primary:** Chapters 4–5 and 8–9 of Wright and Recht (2022), *Optimization for Data Analysis*
- **Theoretical Depth:** Sections 3.7 and 6.1 of Bubeck (2015), *Convex Optimization: Algorithms and Complexity*

12.1 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is arguably the most important algorithm in modern machine learning. It enables training of models on datasets far too large to fit in memory by processing one (or a few) data points at a time.

12.1.1 Motivation: Finite-Sum Problems

Many machine learning problems take the form of **empirical risk minimization**:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m f_i(\mathbf{x}), \quad (12.1)$$

where each $f_i(\mathbf{x})$ represents the loss on the i -th training example.

Example 12.1 (Linear Regression). For least squares regression with data $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$:

$$f_i(\mathbf{x}) = \frac{1}{2}(\mathbf{a}_i^\top \mathbf{x} - b_i)^2.$$

The gradient is $\nabla f_i(\mathbf{x}) = (\mathbf{a}_i^\top \mathbf{x} - b_i)\mathbf{a}_i$.

Example 12.2 (Logistic Regression). For binary classification with labels $b_i \in \{-1, +1\}$:

$$f_i(\mathbf{x}) = \log(1 + e^{-b_i \mathbf{a}_i^\top \mathbf{x}}).$$

The gradient is $\nabla f_i(\mathbf{x}) = -\frac{b_i \mathbf{a}_i}{1 + e^{b_i \mathbf{a}_i^\top \mathbf{x}}}.$

The computational challenge: For gradient descent on problem (12.1), each iteration requires computing the full gradient:

$$\nabla f(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \nabla f_i(\mathbf{x}).$$

When m is large (e.g., millions of training examples), this is prohibitively expensive. Modern datasets can have $m = 10^6$ to 10^9 examples!

12.1.2 The SGD Algorithm

The key insight of SGD is to use a **stochastic estimate** of the gradient instead of the full gradient.

Definition 12.1 (Stochastic Gradient). A **stochastic gradient** at \mathbf{x} is a random vector \mathbf{g} satisfying

$$\mathbb{E}[\mathbf{g}] = \nabla f(\mathbf{x}).$$

That is, the stochastic gradient is an unbiased estimator of the true gradient.

For finite-sum problems (12.1), a simple stochastic gradient is obtained by sampling a single index i uniformly at random:

$$\mathbf{g} = \nabla f_i(\mathbf{x}).$$

Since $\mathbb{E}_i[\nabla f_i(\mathbf{x})] = \frac{1}{m} \sum_{i=1}^m \nabla f_i(\mathbf{x}) = \nabla f(\mathbf{x})$, this is indeed unbiased.

Algorithm: Stochastic Gradient Descent (SGD)

Input: Starting point \mathbf{x}_1 , step size schedule $\{\lambda_k\}_{k=1}^\infty$.

For $k = 1, 2, \dots$:

1. Sample index i_k uniformly at random from $\{1, \dots, m\}$.
2. Compute stochastic gradient: $\mathbf{g}_k = \nabla f_{i_k}(\mathbf{x}_k)$.
3. Update: $\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \mathbf{g}_k$.

Remark 12.1 (Computational Advantage). Each SGD iteration costs $O(n)$ operations (gradient of one f_i), compared to $O(mn)$ for gradient descent. This is an m -fold improvement per iteration!

12.1.3 Step Size Selection

Unlike gradient descent, SGD requires **decreasing step sizes** to converge. The reason is the noise in the stochastic gradient: with a fixed step size, the iterates would oscillate around the optimum forever.

Theorem 12.1 (SGD Step Size Requirements). *For SGD to converge, the step sizes must satisfy the **Robbins-Monro conditions**:*

$$\sum_{k=1}^{\infty} \lambda_k = \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \lambda_k^2 < \infty. \quad (12.2)$$

The first condition ensures we can reach any point in the space; the second ensures the noise diminishes. A common choice satisfying these con-

ditions is:

$$\lambda_k = \frac{c}{k} \quad \text{or} \quad \lambda_k = \frac{c}{\sqrt{k}}$$

for some constant $c > 0$.

12.1.4 Convergence Analysis

Theorem 12.2 (SGD Convergence for Strongly Convex Functions).

Let f be μ -strongly convex and L -smooth, with each f_i having L -Lipschitz gradient. Assume the stochastic gradients have bounded variance: $\mathbb{E}[\|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\|^2] \leq \sigma^2$.

With step size $\lambda_k = \frac{2}{\mu(k+1)}$, SGD satisfies:

$$\mathbb{E}[f(\mathbf{x}_k) - f(\mathbf{x}^*)] \leq \frac{2L\sigma^2}{\mu^2 k}. \quad (12.3)$$

Remark 12.2 (Comparison with Gradient Descent). SGD achieves $O(1/k)$ convergence (in expectation), compared to $O((1 - 1/\kappa)^k)$ for gradient descent on strongly convex problems. However, each SGD iteration is m times cheaper. The total cost to achieve ϵ -accuracy is:

- GD: $O(m \cdot \kappa \log(1/\epsilon))$ gradient computations
- SGD: $O(\sigma^2 / (\mu^2 \epsilon))$ gradient computations

For large m and moderate accuracy ϵ , SGD is often dramatically faster.

12.1.5 Mini-Batch SGD

In practice, rather than using a single sample, we often use a **mini-batch** of B samples to reduce variance.

Algorithm: Mini-Batch SGD

For $k = 1, 2, \dots$:

1. Sample a mini-batch $\mathcal{B}_k \subset \{1, \dots, m\}$ of size $|\mathcal{B}_k| = B$.
2. Compute mini-batch gradient: $\mathbf{g}_k = \frac{1}{B} \sum_{i \in \mathcal{B}_k} \nabla f_i(\mathbf{x}_k)$.
3. Update: $\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \mathbf{g}_k$.

Remark 12.3 (Variance Reduction). The variance of the mini-batch gradient is σ^2/B , so larger batches reduce noise. However, they also increase computation per iteration. Typical batch sizes range from 32 to 512 in deep learning.

12.2 Momentum Methods

Momentum methods accelerate gradient descent by incorporating information from previous iterations. They are particularly effective for ill-conditioned problems.

12.2.1 Polyak's Heavy Ball Method

Definition 12.2 (Heavy Ball Method). The **heavy ball method** (Polyak, 1964) updates:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda \nabla f(\mathbf{x}_k) + \beta(\mathbf{x}_k - \mathbf{x}_{k-1}), \quad (12.4)$$

where $\lambda > 0$ is the step size and $\beta \in [0, 1)$ is the **momentum parameter**.

The term $\beta(\mathbf{x}_k - \mathbf{x}_{k-1})$ is the “momentum”—it pushes the iterate in the direction it was already moving. This helps the algorithm:

- Accelerate through flat regions (where gradients are small)
- Dampen oscillations in steep, narrow valleys

Remark 12.4 (Physical Interpretation). The name “heavy ball” comes from the analogy of a ball rolling down a hill. The momentum term simulates inertia: the ball continues moving even when the slope is zero.

12.2.2 Nesterov's Accelerated Gradient

Nesterov's accelerated gradient (NAG) method, introduced in 1983, is a landmark result in optimization. It achieves the **optimal** convergence rate for smooth convex functions.

Algorithm: Nesterov's Accelerated Gradient (NAG)

Input: Starting point $\mathbf{x}_1 = \mathbf{y}_1$, step size $\lambda = 1/L$.

For $k = 1, 2, \dots$:

1. Compute gradient at extrapolated point:

$$\mathbf{x}_{k+1} = \mathbf{y}_k - \lambda \nabla f(\mathbf{y}_k)$$

2. Update momentum:

$$\mathbf{y}_{k+1} = \mathbf{x}_{k+1} + \frac{k-1}{k+2}(\mathbf{x}_{k+1} - \mathbf{x}_k)$$

The key difference from the heavy ball method is that NAG computes the gradient at an extrapolated point \mathbf{y}_k rather than the current iterate \mathbf{x}_k . This “look-ahead” provides better information about where to go.

12.2.3 Convergence Rates for Accelerated Methods

Theorem 12.3 (Convergence of Nesterov's Method—Convex Case).

Let f be convex and L -smooth. Nesterov's accelerated gradient with step size $\lambda = 1/L$ satisfies:

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2L\|\mathbf{x}_1 - \mathbf{x}^*\|^2}{(k+1)^2} = O\left(\frac{1}{k^2}\right). \quad (12.5)$$

Theorem 12.4 (Convergence of Nesterov's Method—Strongly Convex Case). *Let f be μ -strongly convex and L -smooth. With appropriate parameters, Nesterov's method satisfies:*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq L\|\mathbf{x}_1 - \mathbf{x}^*\|^2 \left(1 - \frac{1}{\sqrt{\kappa}}\right)^k. \quad (12.6)$$

Remark 12.5 (Comparison of Convergence Rates). The following table summarizes convergence rates:

Method	Convex	Strongly Convex
Gradient Descent	$O(1/k)$	$O((1 - 1/\kappa)^k)$
Nesterov Acceleration	$O(1/k^2)$	$O((1 - 1/\sqrt{\kappa})^k)$

For convex functions, acceleration improves the rate from $O(1/k)$ to $O(1/k^2)$ —a quadratic speedup. For strongly convex functions with large condition number κ , the iteration complexity improves from $O(\kappa \log(1/\epsilon))$ to $O(\sqrt{\kappa} \log(1/\epsilon))$ —a $\sqrt{\kappa}$ speedup.

Remark 12.6 (Optimality). Nesterov proved that the $O(1/k^2)$ rate is **optimal** for first-order methods on smooth convex functions. No algorithm using only gradient information can do better in the worst case!

12.2.4 Momentum in Deep Learning

Momentum is ubiquitous in deep learning, though the theoretical guarantees are weaker for nonconvex problems. Common variants include:

- **SGD with Momentum:** Combines SGD with Polyak momentum:

$$\mathbf{v}_{k+1} = \beta \mathbf{v}_k + \mathbf{g}_k, \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \lambda \mathbf{v}_{k+1}.$$

Typical value: $\beta = 0.9$.

- **Adam:** Adaptive learning rates with momentum (Kingma & Ba, 2015). Maintains running averages of both gradients and squared gradients.
- **AdaGrad, RMSProp:** Adaptive step sizes based on gradient history.

12.3 Proximal Gradient Methods

Many modern optimization problems involve **nonsmooth** terms, such as the ℓ_1 regularizer in LASSO. Proximal gradient methods handle such problems elegantly.

12.3.1 Composite Optimization Problems

Consider problems of the form:

$$\min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}), \quad (12.7)$$

where:

- f is smooth (differentiable with Lipschitz gradient)
- g is convex but possibly **nonsmooth**

Example 12.3 (LASSO). The LASSO problem has $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|^2$ (smooth) and $g(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$ (nonsmooth).

Example 12.4 (Constrained Optimization). Constrained problems $\min_{\mathbf{x} \in C} f(\mathbf{x})$ can be written as (12.7) with $g(\mathbf{x}) = I_C(\mathbf{x})$, the indicator function of C :

$$I_C(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

12.3.2 The Proximal Operator

Definition 12.3 (Proximal Operator). The **proximal operator** of a function g with parameter $\lambda > 0$ is:

$$\text{prox}_{\lambda g}(\mathbf{v}) = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\{ g(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{v}\|^2 \right\}. \quad (12.8)$$

Intuitively, the proximal operator finds a point that balances being close to \mathbf{v} with having a small value of g .

Example 12.5 (Soft Thresholding—Proximal of ℓ_1 Norm). For $g(\mathbf{x}) = \|\mathbf{x}\|_1$, the proximal operator is the **soft thresholding** operator:

$$[\text{prox}_{\lambda g}(\mathbf{v})]_j = \operatorname{sign}(v_j) \max\{|v_j| - \lambda, 0\} = \begin{cases} v_j - \lambda & \text{if } v_j > \lambda, \\ 0 & \text{if } |v_j| \leq \lambda, \\ v_j + \lambda & \text{if } v_j < -\lambda. \end{cases}$$

This operator shrinks small components to zero, inducing sparsity.

Derivation of Soft Thresholding. Since $g(\mathbf{x}) = \|\mathbf{x}\|_1 = \sum_{j=1}^n |x_j|$ is separable, the proximal problem decomposes into n independent scalar problems:

$$\text{prox}_{\lambda g}(\mathbf{v}) = \operatorname{argmin}_{\mathbf{x}} \sum_{j=1}^n \left(|x_j| + \frac{1}{2\lambda} (x_j - v_j)^2 \right).$$

Each component x_j^* solves:

$$x_j^* = \operatorname{argmin}_{x \in \mathbb{R}} \left\{ |x| + \frac{1}{2\lambda} (x - v_j)^2 \right\}.$$

Define $h(x) = |x| + \frac{1}{2\lambda} (x - v)^2$ (dropping subscript j for clarity). We consider three cases:

Case 1: $x^* > 0$. Then $|x| = x$ and $h(x) = x + \frac{1}{2\lambda} (x - v)^2$. Setting $h'(x) = 1 + \frac{1}{\lambda} (x - v) = 0$ gives $x^* = v - \lambda$. This is positive only if $v > \lambda$.

Case 2: $x^* < 0$. Then $|x| = -x$ and $h(x) = -x + \frac{1}{2\lambda} (x - v)^2$. Setting $h'(x) = -1 + \frac{1}{\lambda} (x - v) = 0$ gives $x^* = v + \lambda$. This is negative only if $v < -\lambda$.

Case 3: $|v| \leq \lambda$. If $v \in [-\lambda, \lambda]$, then neither Case 1 nor Case 2 applies. We check that $x^* = 0$ is optimal by verifying the subdifferential condition: $0 \in \partial h(0) = [-1, 1] + \frac{1}{\lambda} (0 - v) = [-1 - v/\lambda, 1 - v/\lambda]$. This holds when $|v| \leq \lambda$.

Combining all cases yields the soft thresholding formula. \square

Example 12.6 (Projection—Proximal of Indicator Function). For $g(\mathbf{x}) = I_C(\mathbf{x})$ (indicator of a convex set C):

$$\text{prox}_{\lambda g}(\mathbf{v}) = \Pi_C(\mathbf{v}) = \operatorname{argmin}_{\mathbf{x} \in C} \|\mathbf{x} - \mathbf{v}\|^2.$$

The proximal operator is simply the **projection** onto C , independent of λ .

12.3.3 The Proximal Gradient Algorithm

Algorithm: Proximal Gradient Method

Input: Starting point \mathbf{x}_1 , step size $\lambda = 1/L$ where L is the Lipschitz constant of ∇f .

For $k = 1, 2, \dots$:

1. Gradient step on smooth part: $\mathbf{y}_k = \mathbf{x}_k - \lambda \nabla f(\mathbf{x}_k)$
2. Proximal step on nonsmooth part: $\mathbf{x}_{k+1} = \text{prox}_{\lambda g}(\mathbf{y}_k)$

Equivalently:

$$\mathbf{x}_{k+1} = \text{prox}_{\lambda g}(\mathbf{x}_k - \lambda \nabla f(\mathbf{x}_k)). \quad (12.9)$$

Remark 12.7 (Special Cases). • When $g = 0$: Proximal gradient reduces to standard gradient descent.

- When $g = I_C$: Proximal gradient reduces to **projected gradient descent**:

$$\mathbf{x}_{k+1} = \Pi_C(\mathbf{x}_k - \lambda \nabla f(\mathbf{x}_k)).$$

- When $g = \lambda \|\cdot\|_1$: The algorithm is called **ISTA** (Iterative Shrinkage-Thresholding Algorithm).

12.3.4 Convergence of Proximal Gradient

Theorem 12.5 (Convergence of Proximal Gradient). *Let f be convex and L -smooth, and let g be convex (possibly nonsmooth). The proximal gradient method with step size $\lambda = 1/L$ satisfies:*

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{L\|\mathbf{x}_1 - \mathbf{x}^*\|^2}{2k} = O\left(\frac{1}{k}\right). \quad (12.10)$$

If f is additionally μ -strongly convex, then:

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \left(1 - \frac{\mu}{L}\right)^{k-1} (F(\mathbf{x}_1) - F(\mathbf{x}^*)). \quad (12.11)$$

Proof. The proof mirrors that of gradient descent (Theorem 9.8), with the proximal operator playing a key role.

Step 1: Sufficient decrease per iteration. Define $\mathbf{y}_k = \mathbf{x}_k - \frac{1}{L} \nabla f(\mathbf{x}_k)$ (the gradient step). By L -smoothness of f :

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T (\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{L}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2.$$

Adding $g(\mathbf{x}_{k+1})$ to both sides and using $F = f + g$:

$$F(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T (\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{L}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1}).$$

Step 2: Use the proximal optimality condition. Since $\mathbf{x}_{k+1} = \text{prox}_{g/L}(\mathbf{y}_k)$, by the optimality condition for the proximal problem:

$$\mathbf{0} \in \partial g(\mathbf{x}_{k+1}) + L(\mathbf{x}_{k+1} - \mathbf{y}_k) = \partial g(\mathbf{x}_{k+1}) + L(\mathbf{x}_{k+1} - \mathbf{x}_k) + \nabla f(\mathbf{x}_k).$$

This means there exists $\boldsymbol{\xi} \in \partial g(\mathbf{x}_{k+1})$ such that $\nabla f(\mathbf{x}_k) + L(\mathbf{x}_{k+1} - \mathbf{x}_k) + \boldsymbol{\xi} = \mathbf{0}$.

Step 3: Bound using convexity of g . By convexity of g : $g(\mathbf{x}^*) \geq g(\mathbf{x}_{k+1}) + \boldsymbol{\xi}^T (\mathbf{x}^* - \mathbf{x}_{k+1})$.

$$\text{Therefore: } g(\mathbf{x}_{k+1}) \leq g(\mathbf{x}^*) - \boldsymbol{\xi}^T (\mathbf{x}^* - \mathbf{x}_{k+1}) = g(\mathbf{x}^*) + \boldsymbol{\xi}^T (\mathbf{x}_{k+1} - \mathbf{x}^*).$$

Step 4: Combine the bounds. Substituting into the inequality from Step 1 and using $\boldsymbol{\xi} = -\nabla f(\mathbf{x}_k) - L(\mathbf{x}_{k+1} - \mathbf{x}_k)$:

$$\begin{aligned} F(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T (\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{L}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ &\quad + g(\mathbf{x}^*) + \boldsymbol{\xi}^T (\mathbf{x}_{k+1} - \mathbf{x}^*). \end{aligned}$$

By convexity of f : $f(\mathbf{x}_k) \leq f(\mathbf{x}^*) + \nabla f(\mathbf{x}_k)^T (\mathbf{x}_k - \mathbf{x}^*)$.

After algebraic manipulation (similar to Theorem 9.8):

$$F(\mathbf{x}_{k+1}) - F(\mathbf{x}^*) \leq \frac{L}{2} (\|\mathbf{x}_k - \mathbf{x}^*\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2).$$

Step 5: Telescope. Summing from $k = 1$ to $K - 1$ and using the decreasing property of $F(\mathbf{x}_k)$:

$$(K - 1)(F(\mathbf{x}_K) - F(\mathbf{x}^*)) \leq \sum_{k=1}^{K-1} (F(\mathbf{x}_{k+1}) - F(\mathbf{x}^*)) \leq \frac{L}{2} \|\mathbf{x}_1 - \mathbf{x}^*\|^2.$$

This gives $F(\mathbf{x}_K) - F(\mathbf{x}^*) \leq \frac{L \|\mathbf{x}_1 - \mathbf{x}^*\|^2}{2(K-1)} \leq \frac{L \|\mathbf{x}_1 - \mathbf{x}^*\|^2}{2K}$ for $K \geq 2$.

The strongly convex case follows by applying the PL inequality as in Theorem 9.9. \square

Remark 12.8 (Same Rates as Gradient Descent). Proximal gradient achieves the same convergence rates as gradient descent, even though it can handle nonsmooth regularizers!

12.3.5 Accelerated Proximal Gradient (FISTA)

Just as gradient descent can be accelerated by Nesterov's method, proximal gradient can be accelerated. The resulting algorithm is called **FISTA** (Fast Iterative Shrinkage-Thresholding Algorithm).

Algorithm: FISTA (Accelerated Proximal Gradient)

Input: Starting point $\mathbf{x}_1 = \mathbf{y}_1$, step size $\lambda = 1/L$.

For $k = 1, 2, \dots$:

1. $\mathbf{x}_{k+1} = \text{prox}_{\lambda g}(\mathbf{y}_k - \lambda \nabla f(\mathbf{y}_k))$
2. $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$ (with $t_1 = 1$)
3. $\mathbf{y}_{k+1} = \mathbf{x}_{k+1} + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}_{k+1} - \mathbf{x}_k)$

Theorem 12.6 (Convergence of FISTA). *FISTA achieves the accelerated rate:*

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{2L\|\mathbf{x}_1 - \mathbf{x}^*\|^2}{(k+1)^2} = O\left(\frac{1}{k^2}\right). \quad (12.12)$$

12.4 Summary and Connections

This chapter has introduced three families of modern first-order methods:

1. **Stochastic Gradient Descent:** Essential for large-scale machine learning. Trades slower convergence rate for dramatically cheaper iterations.
2. **Momentum/Accelerated Methods:** Provably faster than gradient descent, achieving optimal rates for first-order methods on smooth convex functions.

3. **Proximal Gradient Methods:** Extend gradient descent to handle nonsmooth regularizers, enabling sparse optimization (LASSO, etc.).

Method	Handles	Cost/Iter	Rate (strongly cvx)
Gradient Descent	Smooth	$O(mn)$	$O((1 - 1/\kappa)^k)$
Nesterov	Smooth	$O(mn)$	$O((1 - 1/\sqrt{\kappa})^k)$
SGD	Smooth	$O(n)$	$O(1/k)$
Proximal Gradient	Smooth + Nonsmooth	$O(mn + n)$	$O((1 - 1/\kappa)^k)$
FISTA	Smooth + Nonsmooth	$O(mn + n)$	$O(1/k^2)$

These methods form the computational foundation for modern machine learning, from training linear models to deep neural networks. Understanding their convergence properties helps practitioners choose the right algorithm for their specific problem.

Recommended Reading

Further Reading:

- Chapter 5 of Wright and Recht (2022) for SGD variants and practical considerations
- Chapter 4 of Wright and Recht (2022) for detailed derivation of momentum methods
- Chapters 8–9 of Wright and Recht (2022) for proximal methods and ADMM
- Section 3.7 of Bubeck (2015) for the theoretical analysis of Nesterov's method

Part IV

Constrained Optimization

Chapter 13

Equality-Constrained Optimization

This chapter introduces optimization problems with equality constraints. We develop the first-order necessary conditions through the Lagrange multiplier theorem, explore various interpretations of these conditions, and examine cases where regularity assumptions fail. We then specialize to convex programs with linear equality constraints, including convex quadratic programs, where the first-order conditions become sufficient for global optimality. The chapter concludes with second-order necessary and sufficient conditions for equality-constrained problems.

Recommended Reading

- Sections 4.1 and 4.2 of Bazaraa, Sherali, and Shetty (2006)
- Chapter 12 of Nocedal and Wright

13.1 Problem Setup and Regularity Assumptions

We now transition from unconstrained optimization to problems involving constraints. This section focuses on optimization problems with equality constraints.

13.1.1 Problem Formulation

Consider optimization problems with **equality constraints** of the form:

$$\begin{aligned} & \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \\ & \text{subject to } \mathbf{h}(\mathbf{x}) = \mathbf{0}, \end{aligned} \tag{13.1}$$

where the constraint $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ is equivalent to

$$h_i(\mathbf{x}) = 0, \quad \forall i \in \{1, \dots, m\}.$$

We assume that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are continuously differentiable, with $m < n$ (fewer constraints than variables).

Definition 13.1 (Feasible Set and Feasible Point). The **feasible set** is defined as

$$X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}.$$

A point $\mathbf{x} \in X$ is called a **feasible point**.

13.1.2 The Jacobian Matrix

Definition 13.2 (Jacobian Matrix). The **Jacobian matrix of \mathbf{h} at \mathbf{x}** is the $m \times n$ matrix

$$J_h(\mathbf{x}) = \begin{pmatrix} \nabla h_1(\mathbf{x})^T \\ \vdots \\ \nabla h_m(\mathbf{x})^T \end{pmatrix},$$

with the (i, j) th entry equal to $J_{h,ij}(\mathbf{x}) = \frac{\partial h_i}{\partial x_j}(\mathbf{x})$.

The i th row of the Jacobian matrix is the transpose of the gradient of the i th constraint function.

13.1.3 Regular Points

Definition 13.3 (Regular Point). A feasible point $\mathbf{x}^* \in X$ is called a **regular point** if

$$\text{rank}(J_h(\mathbf{x}^*)) = m,$$

i.e., if the gradients $\nabla h_1(\mathbf{x}^*), \nabla h_2(\mathbf{x}^*), \dots, \nabla h_m(\mathbf{x}^*)$ are linearly independent.

The regularity condition is also known as the **linear independence constraint qualification (LICQ)**. It ensures that the constraints are locally well-behaved near \mathbf{x}^* .

Example 13.1 (Verifying Regularity). Consider the following constraint functions \mathbf{h} . We verify that all feasible points in $X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$ are regular.

1. $h(\mathbf{x}) = 2x_1^2 + x_2^2 - 1$.

The gradient is $\nabla h(\mathbf{x}) = (4x_1, 2x_2)^T$. This is zero only when $x_1 = x_2 = 0$, but $(0, 0)$ is not feasible since $h(0, 0) = -1 \neq 0$. Therefore, all feasible points are regular.

2. $\mathbf{h}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$, where $A \in \mathbb{R}^{m \times n}$ with $\text{rank}(A) = m < n$ and $\mathbf{b} \in \mathbb{R}^m$.

The Jacobian is $J_h(\mathbf{x}) = A$, which has rank m everywhere. Therefore, all points are regular.

3. $h(\mathbf{x}) = 1 - \mathbf{x}^T P \mathbf{x}$, where $P \in \mathbb{R}^{n \times n}$ is symmetric positive definite.

The gradient is $\nabla h(\mathbf{x}) = -2P\mathbf{x}$. This is zero only when $\mathbf{x} = \mathbf{0}$, but $h(\mathbf{0}) = 1 \neq 0$. Therefore, all feasible points are regular.

13.2 First-Order Necessary Conditions: The Lagrange Multiplier Theorem

The Lagrange multiplier theorem provides necessary conditions for a point to be a local minimizer of an equality-constrained optimization problem.

Theorem 13.1 (Lagrange Multiplier Theorem). *Suppose \mathbf{x}^* is a regular point and a local minimizer of the problem*

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{h}(\mathbf{x}) = \mathbf{0}. \end{aligned}$$

Then, there exists a unique $\boldsymbol{\lambda}^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*) \in \mathbb{R}^m$ such that

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(\mathbf{x}^*) = \mathbf{0}. \quad (13.2)$$

Proof. See Section 3.1.1 of Bertsekas for the complete proof. \square

Remark 13.1. The same conditions are necessary for local maxima of the problem.

13.2.1 The Lagrangian Function

Definition 13.4 (Lagrange Multipliers and Lagrangian Function). The scalars λ_i^* for $i = 1, \dots, m$ in Theorem 13.1 are called **Lagrange multipliers**. The function

$$L : \mathbb{R}^{n+m} \rightarrow \mathbb{R}, \quad L(\mathbf{x}, \boldsymbol{\lambda}) := f(\mathbf{x}) + \sum_{i=1}^m \lambda_i h_i(\mathbf{x})$$

is called the **Lagrangian function** of the problem.

Using the Lagrangian, the first-order necessary conditions can be written compactly as:

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}, \quad (13.3)$$

$$\nabla_{\boldsymbol{\lambda}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}. \quad (13.4)$$

Note that condition (13.3) is equivalent to (13.2), and condition (13.4) is equivalent to $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$ (feasibility).

13.3 Interpretations of the Lagrange Multiplier Theorem

The first-order necessary conditions admit several useful geometric interpretations.

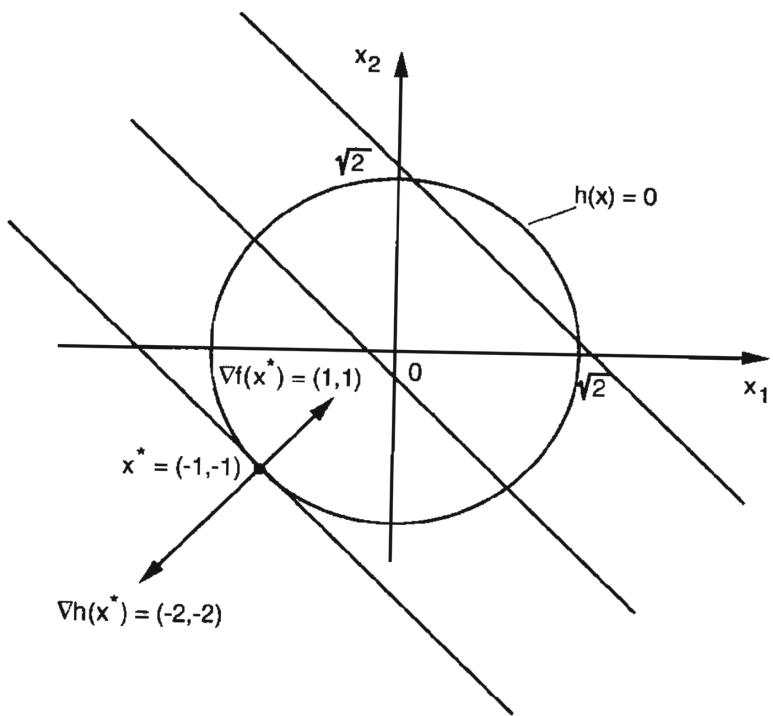


Figure 13.1: Geometric illustration of the Lagrange multiplier conditions. At a local minimum \mathbf{x}^* on the constraint surface $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, the gradient of the objective function $\nabla f(\mathbf{x}^*)$ must lie in the span of the constraint gradients $\nabla h_i(\mathbf{x}^*)$. This ensures that there is no feasible direction along which the objective can decrease.

13.3.1 Gradient in Constraint Gradient Space

Assuming \mathbf{x}^* is a regular point and a local minimum, the condition

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(\mathbf{x}^*) = \mathbf{0}$$

can be rewritten as

$$\nabla f(\mathbf{x}^*) = - \sum_{i=1}^m \lambda_i^* \nabla h_i(\mathbf{x}^*).$$

This shows that $\nabla f(\mathbf{x}^*)$ lies in the span of the constraint gradients:

$$\nabla f(\mathbf{x}^*) \in \text{span}\{\nabla h_1(\mathbf{x}^*), \nabla h_2(\mathbf{x}^*), \dots, \nabla h_m(\mathbf{x}^*)\}.$$

13.3.2 Orthogonality to Feasible Variations

Definition 13.5 (Subspace of First-Order Feasible Variations). The **subspace of first-order feasible variations** at \mathbf{x}^* is defined as

$$V(\mathbf{x}^*) := \{\mathbf{y} \in \mathbb{R}^n : \nabla h_i(\mathbf{x}^*)^T \mathbf{y} = 0, \forall i \in \{1, \dots, m\}\}.$$

This subspace represents directions that are tangent (to first order) to the constraint surface at \mathbf{x}^* . The Lagrange conditions imply that $\nabla f(\mathbf{x}^*)$ is orthogonal to $V(\mathbf{x}^*)$:

$$\nabla f(\mathbf{x}^*)^T \mathbf{y} = 0, \quad \forall \mathbf{y} \in V(\mathbf{x}^*).$$

This is analogous to the condition $\nabla f(\mathbf{x}^*) = \mathbf{0}$ in unconstrained optimization. In the unconstrained case, the gradient must be zero (orthogonal to all directions). In the constrained case, the gradient need only be orthogonal to feasible directions.

13.3.3 Stationary Points of the Lagrangian

The Lagrangian $L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i h_i(\mathbf{x})$ is a function of $n + m$ variables. The first-order necessary conditions

$$\begin{aligned} \nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(\mathbf{x}^*) &= \mathbf{0}, \\ \mathbf{h}(\mathbf{x}^*) &= \mathbf{0} \end{aligned}$$

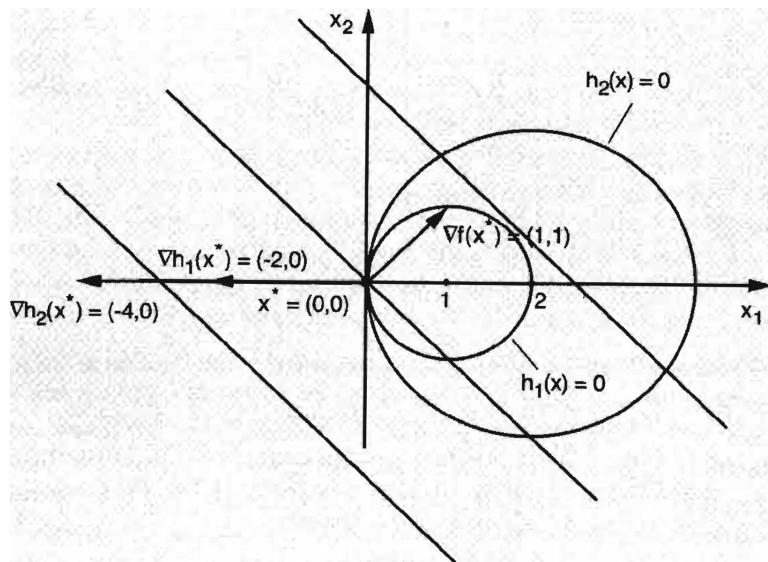


Figure 3.1.2. Illustration of how Lagrange multipliers may not exist

Figure 13.2: Illustration of the orthogonality condition for Lagrange multipliers. The gradient of the objective function $\nabla f(\mathbf{x}^*)$ is orthogonal to the subspace of first-order feasible variations $V(\mathbf{x}^*)$, which is tangent to the constraint surface at \mathbf{x}^* . The constraint gradient $\nabla h(\mathbf{x}^*)$ is normal to this tangent space.

are precisely the conditions that $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is a stationary point of the Lagrangian:

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}, \quad \nabla_{\boldsymbol{\lambda}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}.$$

Remark 13.2. While this observation provides intuition, it is not a proof of the Lagrange multiplier theorem. Finding stationary points of L gives candidates for local minima, but these may also be local maxima or saddle points of the original constrained problem.

13.3.4 Using the First-Order Conditions

The system of $n + m$ equations

$$\begin{aligned} \nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(\mathbf{x}^*) &= \mathbf{0} \quad (n \text{ equations}), \\ \mathbf{h}(\mathbf{x}^*) &= \mathbf{0} \quad (m \text{ equations}) \end{aligned}$$

in $n+m$ unknowns $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ can be solved to find candidate local minimizers. However:

- The system may not be easy to solve in general.
- Solutions to this system are only *candidates* for local minima. They may correspond to local maxima or saddle points.
- Additional conditions (such as second-order conditions) are needed to classify the solutions.

13.4 Numerical Examples

Example 13.2 (Circle Constraint). Consider the problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & x_1 + x_2 \\ \text{subject to} \quad & x_1^2 + x_2^2 - 2 = 0. \end{aligned}$$

Regularity check: The gradient of the constraint is $\nabla h(\mathbf{x}) = (2x_1, 2x_2)^T$, which is nonzero for all feasible points (since $x_1 = x_2 = 0$ is not feasible). Thus, all feasible points are regular.

Lagrangian:

$$L(\mathbf{x}, \lambda) = x_1 + x_2 + \lambda(x_1^2 + x_2^2 - 2).$$

First-order conditions:

$$\begin{aligned}\frac{\partial L}{\partial x_1} &= 1 + 2\lambda^* x_1^* = 0 \implies x_1^* = -\frac{1}{2\lambda^*}, \\ \frac{\partial L}{\partial x_2} &= 1 + 2\lambda^* x_2^* = 0 \implies x_2^* = -\frac{1}{2\lambda^*}, \\ (x_1^*)^2 + (x_2^*)^2 &= 2.\end{aligned}$$

From the first two equations, $x_1^* = x_2^*$. Substituting into the constraint:

$$2(x_1^*)^2 = 2 \implies x_1^* = \pm 1.$$

This gives two candidate solutions:

- $\mathbf{x}^* = (-1, -1)$ with $\lambda^* = \frac{1}{2}$ (this is the global minimum with $f(\mathbf{x}^*) = -2$).
- $\mathbf{x}^* = (1, 1)$ with $\lambda^* = -\frac{1}{2}$ (this is the global maximum with $f(\mathbf{x}^*) = 2$).

Example 13.3 (Ellipse Constraint). Consider the problem

$$\begin{aligned}\min_{\mathbf{x} \in \mathbb{R}^2} \quad & x_1^2 + x_2^2 \\ \text{subject to} \quad & 2x_1^2 + x_2^2 - 1 = 0.\end{aligned}$$

Regularity check: The gradient of the constraint is $\nabla h(\mathbf{x}) = (4x_1, 2x_2)^T$. This is zero only when $x_1 = x_2 = 0$, which is not feasible. Thus, all feasible points are regular.

Lagrangian:

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

First-order conditions:

$$\begin{aligned}2x_1^*(1 + 2\lambda^*) &= 0, \\ 2x_2^*(1 + \lambda^*) &= 0, \\ 2(x_1^*)^2 + (x_2^*)^2 &= 1.\end{aligned}$$

Case analysis:

Case 1: $x_1^* = 0$. Then from the constraint, $(x_2^*)^2 = 1$, so $x_2^* = \pm 1$. From the second equation, $\lambda^* = -1$.

Case 2: $x_2^* = 0$. Then from the constraint, $2(x_1^*)^2 = 1$, so $x_1^* = \pm \frac{1}{\sqrt{2}}$.

From the first equation, $\lambda^* = -\frac{1}{2}$.

This gives four candidate solutions:

1. $\lambda_1^* = -1$, $\mathbf{x}_1^* = (0, 1)$ with $f(\mathbf{x}_1^*) = 1$.
2. $\lambda_2^* = -1$, $\mathbf{x}_2^* = (0, -1)$ with $f(\mathbf{x}_2^*) = 1$.
3. $\lambda_3^* = -\frac{1}{2}$, $\mathbf{x}_3^* = \left(\frac{1}{\sqrt{2}}, 0\right)$ with $f(\mathbf{x}_3^*) = \frac{1}{2}$.
4. $\lambda_4^* = -\frac{1}{2}$, $\mathbf{x}_4^* = \left(-\frac{1}{\sqrt{2}}, 0\right)$ with $f(\mathbf{x}_4^*) = \frac{1}{2}$.

Geometrically, \mathbf{x}_1^* and \mathbf{x}_2^* are global maximizers (farthest from the origin on the ellipse), while \mathbf{x}_3^* and \mathbf{x}_4^* are global minimizers (closest to the origin on the ellipse).

13.5 Failure of Regularity

The regularity assumption in the Lagrange multiplier theorem is essential. When it fails, the first-order conditions may not hold at a local minimum.

Example 13.4 (Non-Regular Point). Consider the problem

$$\begin{aligned} & \min_{\mathbf{x} \in \mathbb{R}^2} x_1 + x_2 \\ & \text{subject to } (x_1 - 1)^2 + x_2^2 - 1 = 0, \\ & \quad (x_1 - 2)^2 + x_2^2 - 4 = 0. \end{aligned}$$

The first constraint describes a circle of radius 1 centered at $(1, 0)$, and the second constraint describes a circle of radius 2 centered at $(2, 0)$. The only point satisfying both constraints is $\mathbf{x}^* = (0, 0)$, which is therefore the unique feasible point and trivially the global minimum.

Checking regularity at $\mathbf{x}^* = (0, 0)$:

$$\begin{aligned} \nabla h_1(\mathbf{x}^*) &= (2(x_1^* - 1), 2x_2^*)^T = (-2, 0)^T, \\ \nabla h_2(\mathbf{x}^*) &= (2(x_1^* - 2), 2x_2^*)^T = (-4, 0)^T. \end{aligned}$$

Since $\nabla h_1(\mathbf{x}^*)$ and $\nabla h_2(\mathbf{x}^*)$ are linearly dependent (both point in the negative x_1 direction), \mathbf{x}^* is **not** a regular point.

Checking the Lagrange conditions: We have $\nabla f(\mathbf{x}^*) = (1, 1)^T$. For the Lagrange conditions to hold, we would need

$$(1, 1)^T = \lambda_1^*(-2, 0)^T + \lambda_2^*(-4, 0)^T = (-2\lambda_1^* - 4\lambda_2^*, 0)^T.$$

This requires $1 = 0$ in the second component, which is impossible. Therefore, no Lagrange multipliers exist, even though $\mathbf{x}^* = (0, 0)$ is the global minimum.

This example demonstrates that the Lagrange multiplier theorem fails when the regularity assumption is violated.

13.6 Convex Programs with Equality Constraints

For convex programs with linear equality constraints, the first-order conditions become both necessary and sufficient for global optimality.

13.6.1 Problem Formulation

Consider **convex problems with linear equality constraints** of the form:

$$\begin{aligned} & \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \\ & \text{subject to } A\mathbf{x} - \mathbf{b} = \mathbf{0}, \end{aligned} \tag{13.5}$$

where f is a **convex** function, $A \in \mathbb{R}^{m \times n}$ (not necessarily full rank), and $\mathbf{b} \in \mathbb{R}^m$.

Remark 13.3. The feasible set $\{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}\}$ is an affine subspace, which is convex. Thus, minimizing a convex function over this set is a convex optimization problem.

Theorem 13.2 (Sufficient Conditions for Convex Programs with Equality Constraints). *Let $\mathbf{x}^* \in \mathbb{R}^n$ and $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ satisfy*

$$\nabla f(\mathbf{x}^*) + A^T \boldsymbol{\lambda}^* = \mathbf{0}, \tag{13.6}$$

$$A\mathbf{x}^* - \mathbf{b} = \mathbf{0}. \tag{13.7}$$

Then \mathbf{x}^ is a **global minimizer** of problem (13.5).*

Proof. By Corollary 2 to Theorem 3.4.3 of Bazaraa et al., a feasible \mathbf{x}^* is optimal if and only if

$$\nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) \geq 0, \quad \forall \mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}.$$

Substituting $\mathbf{v} = \mathbf{x} - \mathbf{x}^*$, this becomes

$$\nabla f(\mathbf{x}^*)^T \mathbf{v} \geq 0, \quad \forall \mathbf{v} \in \mathbb{R}^n : A\mathbf{v} = \mathbf{0}.$$

Since the null space of A is a subspace (closed under negation), this is equivalent to

$$\nabla f(\mathbf{x}^*)^T \mathbf{v} = 0, \quad \forall \mathbf{v} \in \mathbb{R}^n : A\mathbf{v} = \mathbf{0}.$$

This condition holds if and only if $\nabla f(\mathbf{x}^*)$ is in the orthogonal complement of the null space of A , which equals the row space of A . That is, $\nabla f(\mathbf{x}^*) = -A^T \boldsymbol{\lambda}^*$ for some $\boldsymbol{\lambda}^* \in \mathbb{R}^m$, which is precisely condition (13.6). \square

Remark 13.4. The Lagrange multiplier $\boldsymbol{\lambda}^*$ need not be unique if \mathbf{x}^* is not a regular point (i.e., if A does not have full row rank).

13.7 Specialization to Convex Quadratic Programs

We now apply the theory to convex quadratic programs with linear equality constraints.

13.7.1 Problem Formulation

Suppose $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x}$ for some positive definite matrix Q , $A \in \mathbb{R}^{m \times n}$ has full row rank m , and $\mathbf{b} \in \mathbb{R}^m$. Consider the convex program

$$\min_{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}} \frac{1}{2}\mathbf{x}^T Q\mathbf{x}. \quad (13.8)$$

This is a convex program because $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x}$ is convex (its Hessian Q is positive definite) and the feasible set $\{\mathbf{x} : A\mathbf{x} = \mathbf{b}\}$ is convex.

13.7.2 Solving via the Lagrange Conditions

The Lagrangian is

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \boldsymbol{\lambda}^T(A\mathbf{x} - \mathbf{b}).$$

The first-order necessary conditions (which are also sufficient by Theorem 13.2) are:

$$Q\mathbf{x}^* + A^T \boldsymbol{\lambda}^* = \mathbf{0}, \quad (13.9)$$

$$A\mathbf{x}^* - \mathbf{b} = \mathbf{0}. \quad (13.10)$$

Solving the system:

From (13.9): $\mathbf{x}^* = -Q^{-1}A^T \boldsymbol{\lambda}^*$.

Substituting into (13.10):

$$A\mathbf{x}^* = -AQ^{-1}A^T \boldsymbol{\lambda}^* = \mathbf{b}.$$

Therefore:

$$\boldsymbol{\lambda}^* = -(AQ^{-1}A^T)^{-1}\mathbf{b}.$$

The matrix $AQ^{-1}A^T$ is invertible because A has full row rank and Q^{-1} is positive definite.

Substituting back to find \mathbf{x}^* :

$$\mathbf{x}^* = Q^{-1}A^T(AQ^{-1}A^T)^{-1}\mathbf{b}. \quad (13.11)$$

This is the **global minimizer** of problem (13.8).

Remark 13.5. The system (13.9)–(13.10) can be written in matrix form as

$$\begin{pmatrix} Q & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}^* \\ \boldsymbol{\lambda}^* \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix}.$$

This is known as the **KKT system** for the equality-constrained quadratic program.

13.8 Second-Order Optimality Conditions

While first-order conditions identify candidate solutions, second-order conditions help distinguish minima from maxima and saddle points.

13.8.1 The Subspace of First-Order Feasible Variations

Recall the subspace of first-order feasible variations at \mathbf{x}^* :

$$V(\mathbf{x}^*) = \{\mathbf{y} \in \mathbb{R}^n : \nabla h_i(\mathbf{x}^*)^T \mathbf{y} = 0, \forall i \in \{1, \dots, m\}\}.$$

This subspace is the null space of the Jacobian matrix $J_h(\mathbf{x}^*)$.

Example 13.5 (Computing $V(\mathbf{x}^*)$). Consider $h(\mathbf{x}) = x_1^2 + x_2^2 - 2$ with $\mathbf{x}^* = (-1, -1)$.

The gradient is $\nabla h(\mathbf{x}^*) = (2x_1^*, 2x_2^*)^T = (-2, -2)^T$.

The subspace of first-order feasible variations is

$$V(\mathbf{x}^*) = \{\mathbf{y} \in \mathbb{R}^2 : \nabla h(\mathbf{x}^*)^T \mathbf{y} = 0\} = \{\mathbf{y} \in \mathbb{R}^2 : -2y_1 - 2y_2 = 0\} = \{\mathbf{y} \in \mathbb{R}^2 : y_1 + y_2 = 0\}.$$

Example 13.6 (Another Computation). Consider $h(\mathbf{x}) = 2x_1^2 + x_2^2 - 1$ with $\mathbf{x}^* = (0, 1)$.

The gradient is $\nabla h(\mathbf{x}^*) = (4x_1^*, 2x_2^*)^T = (0, 2)^T$.

The subspace of first-order feasible variations is

$$V(\mathbf{x}^*) = \{\mathbf{y} \in \mathbb{R}^2 : 2y_2 = 0\} = \{\mathbf{y} \in \mathbb{R}^2 : y_2 = 0\}.$$

13.8.2 The Hessian of the Lagrangian

For second-order conditions, we assume that f and \mathbf{h} are twice continuously differentiable.

Definition 13.6 (Hessian of the Lagrangian). The **Hessian of the Lagrangian with respect to \mathbf{x}** is

$$\nabla_{\mathbf{x}}^2 L(\mathbf{x}, \boldsymbol{\lambda}) = \nabla^2 f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \nabla^2 h_i(\mathbf{x}).$$

13.8.3 Second-Order Necessary Conditions

Theorem 13.3 (Second-Order Necessary Conditions). Suppose \mathbf{x}^* is a regular point and a local minimizer of the problem

$$\begin{aligned} & \min \quad f(\mathbf{x}) \\ & \text{subject to} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}. \end{aligned}$$

Then, there exists a unique $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ such that

1. $\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$,
2. $\nabla_{\boldsymbol{\lambda}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$,

3. $\mathbf{y}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, \boldsymbol{\lambda}^*) \mathbf{y} \geq 0$ for all $\mathbf{y} \in V(\mathbf{x}^*)$.

Proof. See Section 3.1.1 of Bertsekas for the complete proof. \square

Condition (3) requires the Hessian of the Lagrangian to be positive semidefinite on the subspace of first-order feasible variations. This is analogous to the second-order necessary condition $\nabla^2 f(\mathbf{x}^*) \succeq 0$ for unconstrained problems, but restricted to directions tangent to the constraint surface.

13.8.4 Second-Order Sufficient Conditions

Theorem 13.4 (Second-Order Sufficient Conditions). *Consider the problem $\min_{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}} f(\mathbf{x})$. Suppose $\mathbf{x}^* \in \mathbb{R}^n$ and $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ satisfy*

1. $\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$,
2. $\nabla_{\boldsymbol{\lambda}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$,
3. $\mathbf{y}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, \boldsymbol{\lambda}^*) \mathbf{y} > 0$ for all $\mathbf{y} \in V(\mathbf{x}^*) \setminus \{\mathbf{0}\}$.

Then \mathbf{x}^ is a **strict local minimum** of the problem.*

Proof. See Section 3.2.1 of Bertsekas for the complete proof. \square

Remark 13.6. Unlike the necessary conditions, the sufficient conditions do **not** require \mathbf{x}^* to be a regular point. The strict inequality in condition (3) is the key difference from the necessary conditions.

13.9 Numerical Example of Second-Order Conditions

We revisit Example 13.3 and apply the second-order conditions.

Example 13.7 (Ellipse Constraint Revisited). Consider the problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & x_1^2 + x_2^2 \\ \text{subject to} \quad & 2x_1^2 + x_2^2 - 1 = 0. \end{aligned}$$

From the first-order conditions, we found four candidate solutions:

1. $\lambda_1^* = -1, \mathbf{x}_1^* = (0, 1)$
2. $\lambda_2^* = -1, \mathbf{x}_2^* = (0, -1)$
3. $\lambda_3^* = -\frac{1}{2}, \mathbf{x}_3^* = \left(\frac{1}{\sqrt{2}}, 0\right)$
4. $\lambda_4^* = -\frac{1}{2}, \mathbf{x}_4^* = \left(-\frac{1}{\sqrt{2}}, 0\right)$

Computing the Hessian of the Lagrangian:

The Lagrangian is $L(\mathbf{x}, \lambda) = x_1^2 + x_2^2 + \lambda(2x_1^2 + x_2^2 - 1)$.

The Hessian with respect to \mathbf{x} is

$$\nabla_{\mathbf{x}}^2 L(\mathbf{x}, \lambda) = \begin{pmatrix} 2+4\lambda & 0 \\ 0 & 2+2\lambda \end{pmatrix}.$$

Computing the subspace $V(\mathbf{x}^*)$:

Recall $\nabla h(\mathbf{x}) = (4x_1, 2x_2)^T$.

- $V(\mathbf{x}_1^*) = V(\mathbf{x}_2^*) = \{\mathbf{y} \in \mathbb{R}^2 : 2y_2 = 0\} = \{\mathbf{y} \in \mathbb{R}^2 : y_2 = 0\}$
- $V(\mathbf{x}_3^*) = V(\mathbf{x}_4^*) = \{\mathbf{y} \in \mathbb{R}^2 : 4 \cdot \frac{1}{\sqrt{2}}y_1 = 0\} = \{\mathbf{y} \in \mathbb{R}^2 : y_1 = 0\}$

Evaluating the Hessians:

- $\nabla_{\mathbf{x}}^2 L(\mathbf{x}_1^*, \lambda_1^*) = \nabla_{\mathbf{x}}^2 L(\mathbf{x}_2^*, \lambda_2^*) = \begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix}$
- $\nabla_{\mathbf{x}}^2 L(\mathbf{x}_3^*, \lambda_3^*) = \nabla_{\mathbf{x}}^2 L(\mathbf{x}_4^*, \lambda_4^*) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$

Checking the second-order conditions:

For \mathbf{x}_1^* and \mathbf{x}_2^* with $V(\mathbf{x}^*) = \{\mathbf{y} : y_2 = 0\}$:

$$\mathbf{y}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, \lambda^*) \mathbf{y} = (y_1, 0) \begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ 0 \end{pmatrix} = -2y_1^2 < 0$$

for all $\mathbf{y} \in V(\mathbf{x}^*) \setminus \{\mathbf{0}\}$.

Since the quadratic form is negative, \mathbf{x}_1^* and \mathbf{x}_2^* are **strict local maxima** (not minima).

For \mathbf{x}_3^* and \mathbf{x}_4^* with $V(\mathbf{x}^*) = \{\mathbf{y} : y_1 = 0\}$:

$$\mathbf{y}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, \lambda^*) \mathbf{y} = (0, y_2) \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ y_2 \end{pmatrix} = y_2^2 > 0$$

for all $\mathbf{y} \in V(\mathbf{x}^*) \setminus \{\mathbf{0}\}$.

Since the quadratic form is positive, by Theorem 13.4, \mathbf{x}_3^* and \mathbf{x}_4^* are **strict local minima**.

13.10 Summary

This chapter developed optimality conditions for equality-constrained optimization problems:

1. **First-order necessary conditions (Lagrange multiplier theorem):** At a regular local minimum \mathbf{x}^* , there exist Lagrange multipliers $\boldsymbol{\lambda}^*$ such that the gradient of the Lagrangian vanishes.
2. **Regularity assumption:** The constraint gradients must be linearly independent at \mathbf{x}^* . When this fails, the Lagrange conditions may not hold.
3. **Convex programs with linear equality constraints:** The first-order conditions are both necessary and sufficient for global optimality.
4. **Second-order necessary conditions:** At a regular local minimum, the Hessian of the Lagrangian is positive semidefinite on the subspace of first-order feasible variations.
5. **Second-order sufficient conditions:** If the Hessian of the Lagrangian is positive definite on the subspace of feasible variations, then the point is a strict local minimum.

Chapter 14

Algorithms for Constrained Optimization

This chapter presents fundamental algorithms for solving constrained optimization problems. We begin with an overview of optimization problem classes and available software tools, then develop the projected gradient method for problems with simple constraints. We continue with penalty and augmented Lagrangian methods that transform constrained problems into sequences of unconstrained problems. Finally, we introduce active set methods and interior-point methods, which form the basis for many modern nonlinear programming solvers.

Recommended Reading

- Chapter 15 and Section 16.7 of Nocedal and Wright
- Section 10.5 of Bazaraa, Sherali, and Shetty (2006)
- **Supplementary:** Chapters 7–10 of Wright and Recht (2022) — constrained optimization, proximal methods, duality
- **Interior Point Methods:** Section 5.3 of Bubeck (2015); Chapters 11 and 19 of Boyd and Vandenberghe

14.1 Overview of Optimization Models and Software

We consider the general constrained optimization problem

$$\begin{array}{ll} \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) & \text{(objective function)} \\ \text{s.t. } \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, & \text{(inequality constraints)} \\ \mathbf{h}(\mathbf{x}) = \mathbf{0}. & \text{(equality constraints)} \end{array} \quad (14.1)$$

The structure of the functions f , \mathbf{g} , and \mathbf{h} determines the problem class and the appropriate solution algorithms.

14.1.1 Problem Classes

Definition 14.1 (Linear Programs). A **linear program (LP)** is a problem of the form (14.1) where f , \mathbf{g} , and \mathbf{h} are all **affine functions**. Linear programs are typically solved using the simplex method or interior-point methods.

Definition 14.2 (Convex Programs). A **convex program** is a problem of the form (14.1) where f and each component of \mathbf{g} are **convex functions** and \mathbf{h} is **affine**. The feasible region of a convex program is a convex set, and any local minimizer is also a global minimizer. Convex programs are typically solved using interior-point methods.

Definition 14.3 (Nonconvex Nonlinear Programs). A **nonconvex nonlinear program (NLP)** is a problem of the form (14.1) where f or \mathbf{g} may be **nonconvex** and \mathbf{h} may be **nonlinear**. These problems are generally the most challenging, as they may have multiple local minimizers.

Remark 14.1 (Importance of Problem Structure). In all cases, exploiting problem structure is key to scalability. Special structure such as sparsity, separability, or specific functional forms can lead to dramatic computational speedups.

14.1.2 Modeling Languages and Solvers

Modeling Languages provide high-level interfaces for specifying optimization problems:

- AMPL and GAMS: General-purpose algebraic modeling languages
- Pyomo: Python-based open-source modeling language
- JuMP: Julia-based modeling language with excellent performance
- YALMIP: MATLAB-based toolbox for optimization
- AIMMS: Commercial modeling and optimization platform

Selected Solvers for Convex Programs:

- Commercial: COPT, CPLEX, Gurobi, KNITRO, MOSEK, Xpress
- Non-commercial: Bonmin, Hypatia, Pajarito, SCS, SDPT3, SeDuMi, SHOT

Selected Solvers for Nonconvex NLPs:

- Commercial: CONOPT, KNITRO, MINOS, SNOPT
- Non-commercial: ALGENCAN, FilterSD, FilterSQP, Ipopt, NLopt

14.2 Algorithms for Constrained Optimization: Overview

We consider the problem of minimizing a continuously differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ over a constraint set $X \subseteq \mathbb{R}^n$:

$$\min_{\mathbf{x} \in X} f(\mathbf{x}). \quad (14.2)$$

The goal is to solve (14.2) to local optimality. Starting from an initial guess $\mathbf{x}_1 \in \mathbb{R}^n$, we construct a sequence of iterates $\{\mathbf{x}_k\}_{k=1}^\infty$ and terminate when no progress can be made or an approximate solution has been found.

We assume throughout this chapter that all functions are twice continuously differentiable.

14.2.1 Taxonomy of Algorithms

The main algorithmic approaches for constrained optimization include:

1. **Projected Gradient Method:** Projects iterates onto the feasible set after each gradient step.
2. **Penalty and Augmented Lagrangian Methods:** Transform the constrained problem into a sequence of unconstrained problems.
3. **Sequential Linear Programming (SLP):** Solves a sequence of linear programming approximations.
4. **Sequential Quadratic Programming (SQP):** Solves a sequence of quadratic programming approximations.
5. **Interior-Point Methods:** Maintain strict feasibility with respect to inequality constraints using barrier functions.

14.3 The Projected Gradient Method

The projected gradient method extends gradient descent to constrained problems by projecting iterates onto the feasible set.

14.3.1 Motivation and Basic Framework

Consider the constrained optimization problem

$$\min_{\mathbf{x} \in X} f(\mathbf{x}), \quad (14.3)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable and $X \subset \mathbb{R}^n$ is nonempty and closed.

In unconstrained optimization, we used iterations of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \quad k \geq 1,$$

where \mathbf{d}_k is a descent direction and $\alpha_k > 0$ is a step length. However, in the constrained setting, even if $\mathbf{x}_k \in X$, there is no guarantee that $\mathbf{x}_{k+1} \in X$.

Definition 14.4 (Projection onto a Set). The **projection** of a point

$\mathbf{y} \in \mathbb{R}^n$ onto a set $X \subset \mathbb{R}^n$ is defined as

$$\text{Proj}_X(\mathbf{y}) = \arg \min_{\mathbf{z} \in X} \|\mathbf{z} - \mathbf{y}\|_2.$$

To maintain feasibility, we project onto the set X after each step:

$$\mathbf{x}_{k+1} = \text{Proj}_X(\mathbf{x}_k + \alpha_k \mathbf{d}_k). \quad (14.4)$$

Remark 14.2 (Potential Challenges). Computing the projection can be challenging:

1. If X is nonconvex, $\text{Proj}_X(\mathbf{y})$ may not be unique.
2. Even if $\text{Proj}_X(\mathbf{y})$ is unique, computing it may be as difficult as solving the original optimization problem.

However, for certain special sets X , the projection can be computed efficiently in closed form.

14.4 Computing Projections

14.4.1 Projection for Box Constraints

Box constraints are among the simplest constraint sets for which projections can be computed efficiently.

Definition 14.5 (Box Constraint Set). A **box constraint set** is defined as

$$X = [\mathbf{a}, \mathbf{b}] = \{\mathbf{x} \in \mathbb{R}^n : a_i \leq x_i \leq b_i, \forall i \in \{1, \dots, n\}\},$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ with $a_i \leq b_i$ for all i .

Theorem 14.1 (Projection onto Box Constraints). *For the box constraint set $X = [\mathbf{a}, \mathbf{b}]$ and any $\mathbf{y} \in \mathbb{R}^n$, the projection of \mathbf{y} onto X is*

given componentwise by

$$[\text{Proj}_X(\mathbf{y})]_i = \begin{cases} y_i, & \text{if } a_i \leq y_i \leq b_i, \\ a_i, & \text{if } y_i < a_i, \\ b_i, & \text{if } y_i > b_i, \end{cases} \quad \forall i \in \{1, \dots, n\}.$$

The projection onto a box can be computed in $O(n)$ time, making it extremely efficient.

Example 14.1 (Projection onto a Box in \mathbb{R}^2). Consider the box $X = [0, 1] \times [0, 2]$ and the point $\mathbf{y} = (1.5, -0.5)^T$. The projection is

$$\text{Proj}_X(\mathbf{y}) = (1, 0)^T,$$

since $1.5 > 1$ is clipped to 1 and $-0.5 < 0$ is clipped to 0.

Remark 14.3 (Other Sets with Cheap Projections). Several other convex sets admit efficient projection formulas:

- **Standard simplex:** $X = \{\mathbf{x} \in \mathbb{R}_+^n : \sum_{i=1}^n x_i = 1\}$
- **Halfspaces:** $X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{x} \leq b\}$
- **Unit ball:** $X = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 \leq 1\}$

14.4.2 Projected Gradient Method for Box-Constrained Problems

Consider the box-constrained optimization problem

$$\min_{\mathbf{x} \in [\mathbf{a}, \mathbf{b}]} f(\mathbf{x}),$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable and the box $[\mathbf{a}, \mathbf{b}]$ is nonempty.

Remark 14.4. Since the box $[\mathbf{a}, \mathbf{b}]$ is compact (closed and bounded), and f is continuous, the Weierstrass theorem guarantees that both a minimum and maximum exist.

Definition 14.6 (Projected Gradient Method for Box Constraints). The **projected gradient method** for box-constrained optimization generates iterates according to

$$\mathbf{x}_{k+1} = \text{Proj}_{[\mathbf{a}, \mathbf{b}]}(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)),$$

where $\alpha_k > 0$ is the step length at iteration k .

The step length α_k can be chosen using exact or inexact line search methods, adapted for the projected setting.

14.4.3 Projection for Linear Equality Constraints

Theorem 14.2 (Projection onto Linear Equality Constraints). *Given a matrix $A \in \mathbb{R}^{m \times n}$ of rank $m < n$ and a vector $\mathbf{b} \in \mathbb{R}^m$, consider the affine subspace*

$$X = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}\}.$$

For any $\mathbf{y} \in \mathbb{R}^n$, the projection of \mathbf{y} onto X is given by

$$\text{Proj}_X(\mathbf{y}) = (I_n - A^T(AA^T)^{-1}A)\mathbf{y} + A^T(AA^T)^{-1}\mathbf{b}, \quad (14.5)$$

where I_n is the $n \times n$ identity matrix.

Proof. The projection of \mathbf{y} onto X is the solution to the problem

$$\min_{A\mathbf{z}=\mathbf{b}} \|\mathbf{z} - \mathbf{y}\|_2.$$

Setting $\boldsymbol{\xi} := \mathbf{z} - \mathbf{y}$, we obtain the equivalent problem

$$\min_{A\boldsymbol{\xi}=\mathbf{b}-A\mathbf{y}} \|\boldsymbol{\xi}\|_2,$$

which seeks the minimum-norm solution to the linear system $A\boldsymbol{\xi} = \mathbf{b} - A\mathbf{y}$.

The minimum-norm solution is given by the Moore-Penrose pseudoinverse:

$$\boldsymbol{\xi}^* = A^T(AA^T)^{-1}(\mathbf{b} - A\mathbf{y}).$$

Therefore, the projection is

$$\begin{aligned} \text{Proj}_X(\mathbf{y}) &= \mathbf{y} + \boldsymbol{\xi}^* \\ &= \mathbf{y} + A^T(AA^T)^{-1}(\mathbf{b} - A\mathbf{y}) \\ &= (I_n - A^T(AA^T)^{-1}A)\mathbf{y} + A^T(AA^T)^{-1}\mathbf{b}. \end{aligned} \quad \square$$

Definition 14.7 (Orthogonal Projector onto Nullspace). The matrix

$$P := I_n - A^T(AA^T)^{-1}A$$

is called the **orthogonal projector onto the nullspace** of A . For any $\mathbf{y} \in \mathbb{R}^n$:

$$AP\mathbf{y} = (A - AA^T(AA^T)^{-1}A)\mathbf{y} = \mathbf{0}.$$

Thus, $P\mathbf{y}$ lies in the nullspace $\{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{0}\}$ of A .

14.4.4 Projected Gradient Method for Equality-Constrained Problems

Consider the equality-constrained optimization problem

$$\min_{A\mathbf{x}=\mathbf{b}} f(\mathbf{x}), \quad (14.6)$$

where $A \in \mathbb{R}^{m \times n}$ has rank $m < n$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable.

Theorem 14.3 (Iteration of the Projected Gradient Method). *Starting from a feasible point $\mathbf{x}_k \in X = \{\mathbf{x} : A\mathbf{x} = \mathbf{b}\}$, the projected gradient iteration yields*

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k P\mathbf{d}_k,$$

where $P = I_n - A^T(AA^T)^{-1}A$ and \mathbf{d}_k is the search direction.

Proof. Let $\bar{\mathbf{x}}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$ be the unprojected update. Then

$$\begin{aligned} \mathbf{x}_{k+1} &= \text{Proj}_X(\bar{\mathbf{x}}_{k+1}) \\ &= (I_n - A^T(AA^T)^{-1}A)(\mathbf{x}_k + \alpha_k \mathbf{d}_k) + A^T(AA^T)^{-1}\mathbf{b} \\ &= \mathbf{x}_k - A^T(AA^T)^{-1}A\mathbf{x}_k + \alpha_k P\mathbf{d}_k + A^T(AA^T)^{-1}\mathbf{b} \\ &= \mathbf{x}_k + \alpha_k P\mathbf{d}_k, \end{aligned}$$

where the last equality uses the fact that $A\mathbf{x}_k = \mathbf{b}$ implies $A^T(AA^T)^{-1}A\mathbf{x}_k = A^T(AA^T)^{-1}\mathbf{b}$. \square

Definition 14.8 (Projected Gradient Method for Equality Constraints). The **projected gradient method** for problem (14.6) uses

the iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k P \nabla f(\mathbf{x}_k),$$

where $P = I_n - A^T(AA^T)^{-1}A$ is the nullspace projector.

In other words, instead of taking a step in the direction $-\nabla f(\mathbf{x}_k)$ as in unconstrained gradient descent, we take a step in the direction $-P \nabla f(\mathbf{x}_k)$, which is the projection of the negative gradient onto the nullspace of A .

Remark 14.5 (Choice of Step Length). The step length α_k can be chosen by:

- **Exact line search:** $\alpha_k \in \arg \min_{\alpha \geq 0} f(\mathbf{x}_k - \alpha P \nabla f(\mathbf{x}_k))$
- **Inexact line search:** For example, backtracking line search with Armijo condition

14.5 Penalty Methods

Penalty methods transform constrained optimization problems into sequences of unconstrained problems by penalizing constraint violations in the objective function.

14.5.1 Motivation

Consider the general constrained problem

$$\begin{aligned} & \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \\ \text{s.t. } & g_i(\mathbf{x}) \leq 0, \quad \forall i \in \mathcal{I}, \\ & h_j(\mathbf{x}) = 0, \quad \forall j \in \mathcal{E}. \end{aligned} \tag{14.7}$$

Define the feasible set $X := \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0, \forall i \in \mathcal{I}, h_j(\mathbf{x}) = 0, \forall j \in \mathcal{E}\}$.

For nonconvex problems, even finding a feasible point $\mathbf{x} \in X$ may be difficult. Penalty methods address this by penalizing constraint violations, allowing us to use unconstrained optimization techniques.

14.5.2 Quadratic Penalty Method

We first consider equality-constrained problems:

$$\begin{aligned} & \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \\ & \text{s.t. } h_j(\mathbf{x}) = 0, \quad \forall j \in \mathcal{E}. \end{aligned} \quad (14.8)$$

Definition 14.9 (Quadratic Penalty Function). The **quadratic penalty function** for problem (14.8) is

$$Q(\mathbf{x}; \mu) := f(\mathbf{x}) + \frac{\mu}{2} \sum_{j \in \mathcal{E}} (h_j(\mathbf{x}))^2, \quad (14.9)$$

where $\mu > 0$ is the penalty parameter.

Remark 14.6 (Properties of the Quadratic Penalty). 1. At any feasible point \mathbf{x}^* (where $h_j(\mathbf{x}^*) = 0$ for all $j \in \mathcal{E}$), we have $Q(\mathbf{x}^*; \mu) = f(\mathbf{x}^*)$.
 2. The function $Q(\mathbf{x}; \mu)$ can be minimized using any unconstrained optimization algorithm.
 3. As $\mu \rightarrow +\infty$, global minimizers of $Q(\mathbf{x}; \mu)$ converge to global minimizers of the original problem (14.8).

Theorem 14.4 (Convergence of Quadratic Penalty Method). Let $\{\mu_k\}$ be a sequence of penalty parameters with $\mu_k \rightarrow +\infty$, and let \mathbf{x}_k be a global minimizer of $Q(\mathbf{x}; \mu_k)$ for each k . Then every limit point of $\{\mathbf{x}_k\}$ is a global solution of the equality-constrained problem (14.8).

Proof. See Theorems 17.1 and 17.2 of Nocedal and Wright. \square

14.5.3 Nonsmooth (Exact) Penalty Method

An alternative to the quadratic penalty uses the ℓ_1 norm of constraint violations.

Definition 14.10 (Nonsmooth Penalty Function). The **nonsmooth**

(exact) penalty function for problem (14.8) is

$$\phi(\mathbf{x}; \mu) := f(\mathbf{x}) + \mu \sum_{j \in \mathcal{E}} |h_j(\mathbf{x})|, \quad (14.10)$$

where $\mu > 0$ is the penalty parameter.

Theorem 14.5 (Exactness Property). *Under appropriate regularity conditions, there exists $\bar{\mu} > 0$ such that for all $\mu \geq \bar{\mu}$, any local minimizer of problem (14.8) is also a local minimizer of the penalized problem $\min_{\mathbf{x}} \phi(\mathbf{x}; \mu)$.*

This “exactness” property means that we do not need to drive $\mu \rightarrow \infty$; a sufficiently large but finite μ suffices. However, the function $\phi(\mathbf{x}; \mu)$ is nonsmooth, requiring specialized optimization algorithms.

14.5.4 Penalty Methods for General Constraints

For problems with both equality and inequality constraints, the nonsmooth penalty function becomes

$$\phi(\mathbf{x}; \mu) := f(\mathbf{x}) + \mu \sum_{i \in \mathcal{I}} \max\{0, g_i(\mathbf{x})\} + \mu \sum_{j \in \mathcal{E}} |h_j(\mathbf{x})|. \quad (14.11)$$

The term $\max\{0, g_i(\mathbf{x})\}$ penalizes only violated inequality constraints (those where $g_i(\mathbf{x}) > 0$).

14.6 Augmented Lagrangian Methods

Augmented Lagrangian methods combine the Lagrangian relaxation with a quadratic penalty term, offering advantages over pure penalty methods.

14.6.1 The Augmented Lagrangian Function

Consider the equality-constrained problem (14.8).

Definition 14.11 (Augmented Lagrangian). The **augmented La-**

grangian function is

$$\mathcal{L}_A(\mathbf{x}, \boldsymbol{\lambda}; \mu) := f(\mathbf{x}) - \sum_{j \in \mathcal{E}} \lambda_j h_j(\mathbf{x}) + \frac{\mu}{2} \sum_{j \in \mathcal{E}} (h_j(\mathbf{x}))^2, \quad (14.12)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{E}|}$ is a vector of Lagrange multiplier estimates and $\mu > 0$ is the penalty parameter.

The augmented Lagrangian combines:

- The standard Lagrangian: $f(\mathbf{x}) - \sum_{j \in \mathcal{E}} \lambda_j h_j(\mathbf{x})$
- The quadratic penalty: $\frac{\mu}{2} \sum_{j \in \mathcal{E}} (h_j(\mathbf{x}))^2$

14.6.2 The Augmented Lagrangian Algorithm

Definition 14.12 (Augmented Lagrangian Method). The **augmented Lagrangian method** proceeds as follows:

1. **Initialize:** Choose starting values $\boldsymbol{\lambda}^0$ and $\mu_0 > 0$.
2. **For** $k = 0, 1, 2, \dots$:
 - (a) Approximately minimize $\mathcal{L}_A(\mathbf{x}, \boldsymbol{\lambda}^k; \mu_k)$ with respect to \mathbf{x} to obtain \mathbf{x}^{k+1} .
 - (b) Update the multiplier estimates:
$$\lambda_j^{k+1} = \lambda_j^k - \mu_k h_j(\mathbf{x}^{k+1}), \quad \forall j \in \mathcal{E}.$$
 - (c) Update the penalty parameter μ_{k+1} (increase if constraint violation is not sufficiently reduced).
3. **Terminate** when convergence criteria are satisfied.

Remark 14.7 (Advantages over Pure Penalty Methods). The augmented Lagrangian method has a key advantage: we do not need to increase μ to infinity. By simultaneously updating the Lagrange multiplier estimates, the method can converge with a bounded penalty parameter, avoiding the severe ill-conditioning that arises in pure penalty methods as $\mu \rightarrow \infty$.

Theorem 14.6 (Convergence of Augmented Lagrangian Method). *Under appropriate assumptions, the augmented Lagrangian method converges to a KKT point of the original problem. Moreover, the sequence $\{\mu_k\}$ remains bounded.*

Proof. See Theorems 17.5 and 17.6 of Nocedal and Wright. \square

14.7 Active Set Methods

Active set methods maintain and update estimates of which inequality constraints are active (binding) at the solution.

14.7.1 Active Constraints and the Active Set

Definition 14.13 (Active Constraint). A constraint is said to be **active** (or **binding**) at a feasible point \mathbf{x} if it holds with equality.

Definition 14.14 (Active Set). The **active set** at a point $\mathbf{x} \in \mathbb{R}^n$ is

$$\mathcal{A}(\mathbf{x}) := \{i \in \mathcal{I} : g_i(\mathbf{x}) = 0\} \cup \mathcal{E}.$$

14.7.2 Key Idea

If we knew the active set $\mathcal{A}(\mathbf{x}^*)$ at an optimal solution \mathbf{x}^* , we could solve the simpler equality-constrained problem:

$$\begin{aligned} & \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \\ & \text{s.t. } g_i(\mathbf{x}) = 0, \quad \forall i \in \mathcal{A}(\mathbf{x}^*), \\ & \quad h_j(\mathbf{x}) = 0, \quad \forall j \in \mathcal{E}. \end{aligned} \tag{14.13}$$

Active set methods start with a guess for $\mathcal{A}(\mathbf{x}^*)$ and iteratively update this guess until convergence to a local solution.

Remark 14.8. The simplex method for linear programming is an active set method! At each iteration, it maintains a basis that corresponds to a working set of active constraints.

14.7.3 Sequential Linear Programming (SLP)

Sequential linear programming approximates the nonlinear problem by a sequence of linear programs.

Definition 14.15 (SLP Subproblem). At iteration k , given the current iterate \mathbf{x}_k , SLP solves the linear programming subproblem:

$$\begin{aligned} \min_{\mathbf{p} \in \mathbb{R}^n} & f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} \\ \text{s.t. } & g_i(\mathbf{x}_k) + \nabla g_i(\mathbf{x}_k)^T \mathbf{p} \leq 0, \quad \forall i \in \mathcal{I}, \\ & h_j(\mathbf{x}_k) + \nabla h_j(\mathbf{x}_k)^T \mathbf{p} = 0, \quad \forall j \in \mathcal{E}, \\ & \|\mathbf{p}\|_\infty \leq \Delta_k. \end{aligned} \quad (14.14)$$

The next iterate is $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$, where \mathbf{p}_k solves (14.14).

The constraint $\|\mathbf{p}\|_\infty \leq \Delta_k$ is a trust region constraint that limits the step size, ensuring that the linear approximation remains valid.

14.7.4 Sequential Quadratic Programming (SQP)

Sequential quadratic programming extends SLP by using second-order information.

Definition 14.16 (SQP Subproblem). At iteration k , given the current iterate \mathbf{x}_k , SQP solves the quadratic programming subproblem:

$$\begin{aligned} \min_{\mathbf{p} \in \mathbb{R}^n} & f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}_k, \boldsymbol{\lambda}_k) \mathbf{p} \\ \text{s.t. } & g_i(\mathbf{x}_k) + \nabla g_i(\mathbf{x}_k)^T \mathbf{p} \leq 0, \quad \forall i \in \mathcal{I}, \\ & h_j(\mathbf{x}_k) + \nabla h_j(\mathbf{x}_k)^T \mathbf{p} = 0, \quad \forall j \in \mathcal{E}, \\ & \|\mathbf{p}\|_\infty \leq \Delta_k, \end{aligned} \quad (14.15)$$

where the Lagrangian is $L(\mathbf{x}, \boldsymbol{\lambda}) := f(\mathbf{x}) + \sum_{i \in \mathcal{I}} \lambda_i g_i(\mathbf{x}) + \sum_{j \in \mathcal{E}} \lambda_j h_j(\mathbf{x})$.

SQP is motivated by Newton's method for computing a stationary point of the Lagrangian. The next iterate is $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$, where \mathbf{p}_k solves (14.15).

14.8 Interior-Point Methods

Interior-point methods maintain strict feasibility with respect to inequality constraints throughout the optimization process.

14.8.1 The Barrier Approach

Consider the general constrained problem (14.7). Suppose we have an initial point \mathbf{x}_1 with $g_i(\mathbf{x}_1) < 0$ for all $i \in \mathcal{I}$ (i.e., strictly feasible with respect to the inequality constraints).

Definition 14.17 (Barrier Subproblem). The **logarithmic barrier subproblem** for parameter $\mu > 0$ is

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) - \mu \sum_{i \in \mathcal{I}} \ln(-g_i(\mathbf{x})) \\ \text{s.t.} \quad & h_j(\mathbf{x}) = 0, \quad \forall j \in \mathcal{E}. \end{aligned} \tag{14.16}$$

The logarithmic barrier term $-\mu \sum_{i \in \mathcal{I}} \ln(-g_i(\mathbf{x}))$ has the following properties:

- It is well-defined only when $g_i(\mathbf{x}) < 0$ for all $i \in \mathcal{I}$ (strict feasibility).
- It goes to $+\infty$ as any $g_i(\mathbf{x}) \rightarrow 0^-$ (approaching the boundary from the interior).
- As $\mu \rightarrow 0^+$, its influence diminishes.

14.8.2 Interpretation

Interior-point methods can be understood from two perspectives:

1. **KKT approximation:** The method approximately solves the KKT conditions for the original problem, with μ controlling the accuracy.
2. **Barrier function:** The method maintains strict feasibility by using a barrier function that prevents iterates from reaching the boundary of the feasible region.

14.8.3 Convergence

Theorem 14.7 (Convergence of Interior-Point Methods). *As $\mu \rightarrow 0^+$, local solutions of the barrier subproblem (14.16) converge to local solutions of the original problem (14.7).*

Proof. See Theorem 19.1 of Nocedal and Wright. \square

In practice, interior-point methods solve a sequence of barrier subproblems for decreasing values of μ , using the solution from one subproblem as a warm start for the next.

14.9 Summary and Further Reading

This chapter has presented several fundamental algorithmic approaches for constrained optimization:

1. **Projected Gradient Methods** are simple and effective when projections can be computed cheaply (e.g., box constraints, linear equality constraints).
2. **Penalty Methods** transform constrained problems into unconstrained ones, but may suffer from ill-conditioning as the penalty parameter grows.
3. **Augmented Lagrangian Methods** combine penalty terms with Lagrange multiplier updates, avoiding the ill-conditioning of pure penalty methods.
4. **Active Set Methods** (including SLP and SQP) maintain estimates of the active constraint set and solve sequences of simpler subproblems.
5. **Interior-Point Methods** maintain strict feasibility using barrier functions and are particularly effective for convex programs.

Remark 14.9 (Topics for Further Study). The following topics, while beyond the scope of these notes, are important for understanding modern optimization software:

- Large-scale unconstrained optimization
- Automatic differentiation for computing derivatives

- Derivative-free optimization methods
- Specialization to least-squares problems
- Algorithms for systems of nonlinear equations
- Primal-dual interior-point methods
- Specialized algorithms for quadratic programs

See Chapters 7–11, 14, 16, and 19 of Nocedal and Wright for detailed treatments of these topics.

Chapter 15

KKT Conditions for Inequality-Constrained Problems

This chapter develops the Karush-Kuhn-Tucker (KKT) optimality conditions for nonlinear optimization problems with inequality constraints. We begin by establishing geometric necessary conditions based on cones of feasible and improving directions, then derive the Fritz John necessary conditions, and finally present the KKT conditions along with various constraint qualifications that ensure their validity. The chapter concludes with applications to linear programming and convex optimization.

Recommended Reading

- Section 4.2 of Bazaraa, Sherali, and Shetty (2006)
- Chapter 12 of Nocedal and Wright

15.1 Geometric Necessary Conditions for Constrained Problems

Consider the general constrained optimization problem

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

where $S \subseteq \mathbb{R}^n$ is a nonempty feasible set and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable. Our goal is to characterize what conditions an optimal solution $\mathbf{x}^* \in S$ must

satisfy.

The key insight underlying optimality conditions for constrained problems is that **directions that maintain feasibility should not be descent directions** at an optimal solution. We formalize this geometric notion through the following definitions.

Remark 15.1. Throughout this chapter, we assume that all functions (objective and constraints) are differentiable unless otherwise stated.

15.1.1 Cones of Feasible and Improving Directions

Definition 15.1 (Cone of Feasible Directions). Let $S \subseteq \mathbb{R}^n$ be nonempty and let $\mathbf{x}^* \in \text{cl}(S)$. The **cone of feasible directions** of S at \mathbf{x}^* is

$$D := \{\mathbf{d} \in \mathbb{R}^n : \mathbf{d} \neq \mathbf{0}, \mathbf{x}^* + \lambda\mathbf{d} \in S \text{ for all } \lambda \in (0, \delta) \text{ for some } \delta > 0\}.$$

The cone D contains all nonzero directions along which one can move from \mathbf{x}^* while remaining feasible, at least for sufficiently small step sizes.

Definition 15.2 (Cone of Improving Directions). Given $f : \mathbb{R}^n \rightarrow \mathbb{R}$, the **cone of improving directions** (or **descent directions**) at \mathbf{x}^* is

$$F := \{\mathbf{d} \in \mathbb{R}^n : f(\mathbf{x}^* + \lambda\mathbf{d}) < f(\mathbf{x}^*) \text{ for all } \lambda \in (0, \delta) \text{ for some } \delta > 0\}.$$

Each $\mathbf{d} \in F$ is called an **improving direction** or **descent direction** of f at \mathbf{x}^* .

The fundamental observation is that at an optimal solution, we cannot have a direction that is both feasible and improving.

Remark 15.2. At an optimal solution \mathbf{x}^* , we must have $F \cap D = \emptyset$. However, this geometric condition is difficult to use directly because both F and D are typically hard to compute explicitly.

To make the optimality conditions tractable, we introduce first-order

approximations to these cones. Define

$$\begin{aligned} F_0 &:= \{\mathbf{d} \in \mathbb{R}^n : \nabla f(\mathbf{x}^*)^T \mathbf{d} < 0\}, \\ F'_0 &:= \{\mathbf{d} \in \mathbb{R}^n : \mathbf{d} \neq \mathbf{0}, \nabla f(\mathbf{x}^*)^T \mathbf{d} \leq 0\}. \end{aligned}$$

The set F_0 is the open half-space of directions making an acute angle with $-\nabla f(\mathbf{x}^*)$, while F'_0 is its closure (excluding the origin). These sets relate to the true cone of improving directions as follows:

$$F_0 \subseteq F \subseteq F'_0.$$

15.1.2 Geometric Necessary Condition

Theorem 15.1 (Geometric Necessary Condition for Constrained Problems). *Given a nonempty set $S \subseteq \mathbb{R}^n$ and a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, consider the constrained problem $\min\{f(\mathbf{x}) : \mathbf{x} \in S\}$. If $\mathbf{x}^* \in S$ is a local minimum and f is differentiable at \mathbf{x}^* , then*

$$F_0 \cap D = \emptyset,$$

where $F_0 = \{\mathbf{d} \in \mathbb{R}^n : \nabla f(\mathbf{x}^*)^T \mathbf{d} < 0\}$ and D is the cone of feasible directions at \mathbf{x}^* .

This theorem states that if \mathbf{x}^* is a local minimum, then no feasible direction can be a first-order descent direction. Geometrically, this means the negative gradient $-\nabla f(\mathbf{x}^*)$ cannot point into the interior of the cone of feasible directions.

15.2 Problems with Inequality Constraints

We now specialize to problems with explicit inequality constraints. Consider the feasible set

$$S := \{\mathbf{x} \in X : g_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}\},$$

where $X \subseteq \mathbb{R}^n$ is a nonempty **open** set and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are constraint functions.

Definition 15.3 (Active/Binding Constraints). Given a feasible point

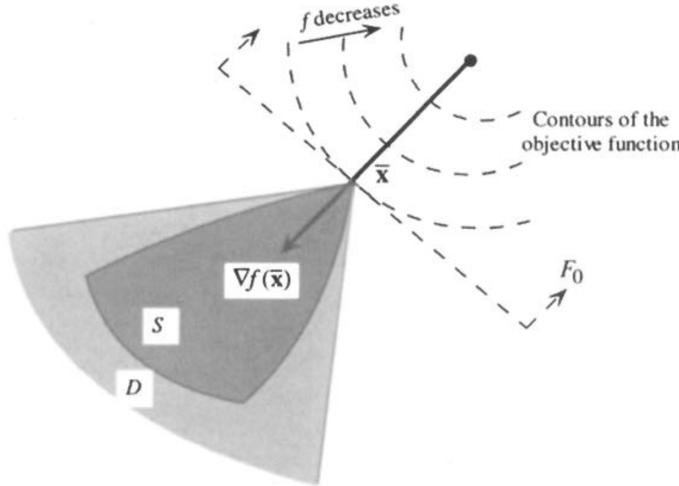


Figure 4.3 Necessary condition $F_0 \cap D = \emptyset$.

Figure 15.1: Illustration of the cone of feasible directions D and the cone of improving directions F_0 at different points in a feasible region. At a local minimum, these cones must have empty intersection.

$\mathbf{x}^* \in S$, the **active set** (or **binding constraint set**) is

$$I := \{i \in \{1, \dots, m\} : g_i(\mathbf{x}^*) = 0\}.$$

Constraints with indices in I are called **active**, **binding**, or **tight** at \mathbf{x}^* .

The active constraints are those that “touch” the boundary of the feasible region at \mathbf{x}^* . Inactive constraints (with $g_i(\mathbf{x}^*) < 0$) do not affect the local geometry near \mathbf{x}^* .

We define first-order approximations to the cone of feasible directions based on the active constraints:

$$\begin{aligned} G_0 &:= \{\mathbf{d} \in \mathbb{R}^n : \nabla g_i(\mathbf{x}^*)^T \mathbf{d} < 0 \text{ for each } i \in I\}, \\ G'_0 &:= \{\mathbf{d} \in \mathbb{R}^n : \mathbf{d} \neq \mathbf{0}, \nabla g_i(\mathbf{x}^*)^T \mathbf{d} \leq 0 \text{ for each } i \in I\}. \end{aligned}$$

Lemma 15.2. *For the feasible set S defined above, we have $G_0 \subseteq D \subseteq G'_0$.*

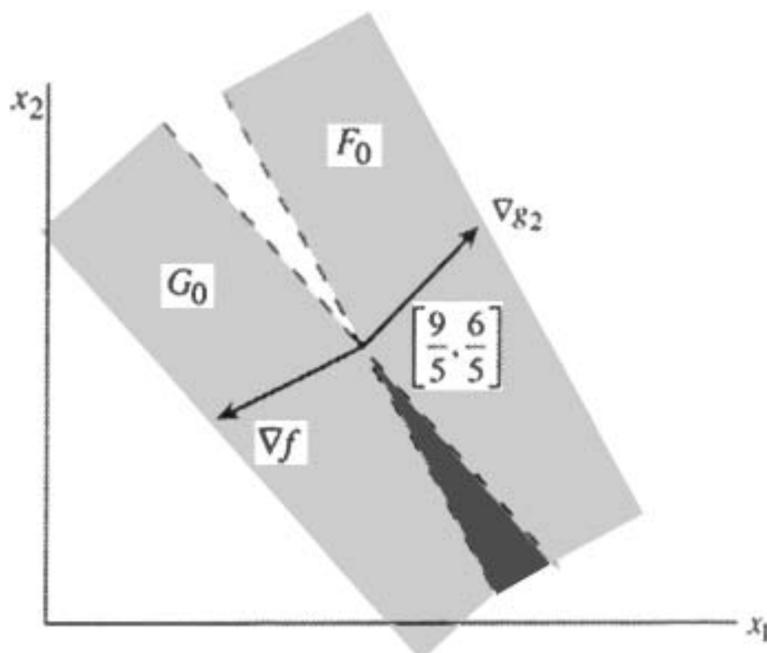


Figure 15.2: Feasible region defined by multiple inequality constraints. The shaded region represents the set $S = \{\mathbf{x} : g_i(\mathbf{x}) \leq 0, i = 1, \dots, m\}$. At any feasible point, the active constraints determine the local geometry.

This lemma shows that G_0 provides an inner approximation to the cone of feasible directions, while G'_0 provides an outer approximation.

15.2.1 Necessary Condition for Inequality-Constrained Problems

Theorem 15.3 (Necessary Condition for Inequality-Constrained Problems). *Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, consider the constrained problem*

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad \forall i \in \{1, \dots, m\}, \\ & \mathbf{x} \in X. \end{aligned}$$

If \mathbf{x}^* is a local minimum, then $F_0 \cap G_0 = \emptyset$, where

- $F_0 = \{\mathbf{d} \in \mathbb{R}^n : \nabla f(\mathbf{x}^*)^T \mathbf{d} < 0\}$,
- $G_0 = \{\mathbf{d} \in \mathbb{R}^n : \nabla g_i(\mathbf{x}^*)^T \mathbf{d} < 0 \text{ for each } i \in I\}$.

Remark 15.3. The condition $F_0 \cap D = \emptyset$ at a local minimum does **not** imply that the point is a local minimum—the converse is not necessarily true. For example, consider $f(\mathbf{x}) = x_2$, $S = \{\mathbf{x} \in \mathbb{R}^2 : x_2 = x_1^2\}$, and $\mathbf{x}^* = (1, 1)$. At this point, $F_0 \cap D = \emptyset$, but \mathbf{x}^* is not a local minimum.

15.2.2 Numerical Example

Example 15.1. Consider the optimization problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & (x_1 - 3)^2 + (x_2 - 2)^2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 \leq 5, \\ & x_1 + x_2 \leq 3, \\ & x_1, x_2 \geq 0. \end{aligned}$$

The objective function measures the squared distance to the point $(3, 2)$, and we seek the feasible point closest to this target.

At $\mathbf{x} = (9/5, 6/5)$: This point lies on the constraint $x_1^2 + x_2^2 = 5$ but is not optimal. We can verify that $F_0 \cap G_0 \neq \emptyset$, meaning there exists a

direction that is both feasible and improving.

At $\mathbf{x}^* = (2, 1)$: This is the optimal solution. The active constraints are $x_1^2 + x_2^2 = 5$ and $x_1 + x_2 = 3$. We have $F_0 \cap G_0 = \emptyset$, confirming the necessary condition.

15.2.3 Issues with the Geometric Necessary Condition

The condition $F_0 \cap G_0 = \emptyset$ can be satisfied trivially in certain degenerate cases:

1. If $\nabla f(\mathbf{x}^*) = \mathbf{0}$, then $F_0 = \emptyset$, so $F_0 \cap G_0 = \emptyset$ trivially.
2. If $\nabla g_i(\mathbf{x}^*) = \mathbf{0}$ for some $i \in I$, then $G_0 = \emptyset$, so $F_0 \cap G_0 = \emptyset$ trivially.
3. The usefulness of the condition depends on how the constraints are represented.

Example 15.2. Consider the problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & (x_1 - 1)^2 + (x_2 - 1)^2 \\ \text{s.t.} \quad & (x_1 + x_2 - 1)^3 \leq 0, \\ & x_1, x_2 \geq 0. \end{aligned}$$

For each \mathbf{x}^* with $x_1 + x_2 = 1$, we have $\nabla g_1(\mathbf{x}^*) = 3(x_1 + x_2 - 1)^2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \mathbf{0}$. Therefore, $G_0 = \emptyset$ and $F_0 \cap G_0 = \emptyset$ for **every** point on the line $x_1 + x_2 = 1$!

However, if we equivalently write the constraint as $x_1 + x_2 - 1 \leq 0$, then $F_0 \cap G_0 = \emptyset$ holds only at $\mathbf{x}^* = (0.5, 0.5)$, which is the true optimal solution.

This example illustrates that the algebraic form of the constraints matters for the optimality conditions. This motivates the development of more robust conditions.

15.3 Fritz John Necessary Optimality Conditions

The geometric condition $F_0 \cap G_0 = \emptyset$ can be converted into an algebraic condition involving the gradients of the objective function and active constraints.

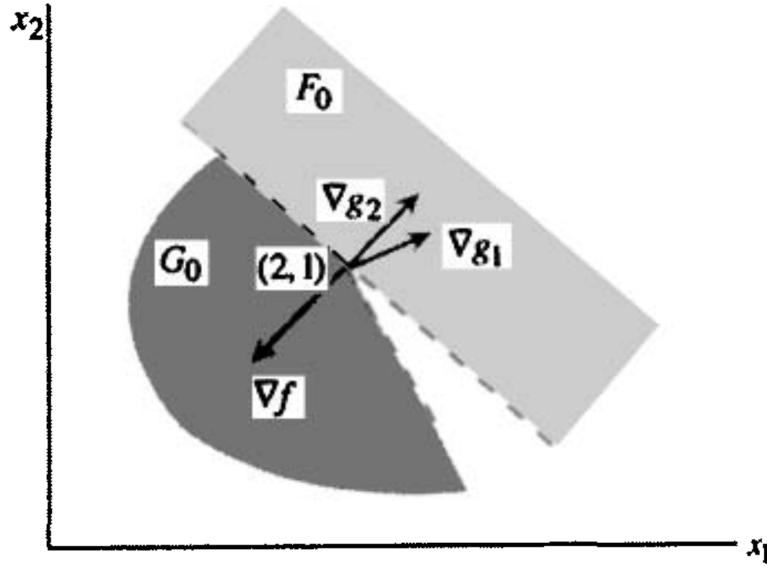


Figure 15.3: Geometric illustration of the feasible region and constraint boundaries. The form in which constraints are written can affect whether the geometric necessary condition correctly identifies the optimal solution.

Theorem 15.4 (Fritz John Necessary Conditions). *Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, consider the constrained problem*

$$\min\{f(\mathbf{x}) : g_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}, \mathbf{x} \in X\}.$$

If \mathbf{x}^ is a local minimum, then there exist scalars u_0 and u_i , $i \in \{1, \dots, m\}$, such that*

$$u_0 \nabla f(\mathbf{x}^*) + \sum_{i=1}^m u_i \nabla g_i(\mathbf{x}^*) = \mathbf{0}, \quad (15.1)$$

$$u_i g_i(\mathbf{x}^*) = 0, \quad \forall i \in \{1, \dots, m\}, \quad (15.2)$$

$$u_0, u_i \geq 0, \quad \forall i \in \{1, \dots, m\}, \quad (15.3)$$

$$(u_0, u_1, \dots, u_m) \neq \mathbf{0}. \quad (15.4)$$

15.3.1 Terminology for Fritz John Conditions

Definition 15.4 (Fritz John Optimality Conditions). The conditions in Theorem 15.4 consist of the following components:

- **Lagrange Multipliers:** The scalars u_0 and $u_i, i \in \{1, \dots, m\}$.
- **Primal Feasibility:** $\mathbf{x}^* \in X$ and $g_i(\mathbf{x}^*) \leq 0$ for all $i \in \{1, \dots, m\}$.
- **Dual Feasibility:** Equations (15.1), (15.3), and (15.4).
- **Complementary Slackness:** Equation (15.2), which states $u_i g_i(\mathbf{x}^*) = 0$ for each i .

Together, these conditions are called the **Fritz John (FJ) Optimality Conditions**.

Definition 15.5 (Fritz John Point). A feasible point \mathbf{x}^* is called a **Fritz John point** (or **FJ point**) if there exist Lagrange multipliers $(u_0^*, u_1^*, \dots, u_m^*)$ such that $(\mathbf{x}^*, u_0^*, \dots, u_m^*)$ satisfies the Fritz John conditions.

Remark 15.4. Complementary slackness (15.2) implies that for each constraint, either $u_i = 0$ (the multiplier is zero) or $g_i(\mathbf{x}^*) = 0$ (the constraint is active). This means inactive constraints have zero multipliers, so we can equivalently state the FJ conditions using only the active constraints:

$$u_0 \nabla f(\mathbf{x}^*) + \sum_{i \in I} u_i \nabla g_i(\mathbf{x}^*) = \mathbf{0}.$$

Remark 15.5 (Necessary Conditions for Local Maxima). For maximization problems $\max f(\mathbf{x})$ subject to $g_i(\mathbf{x}) \leq 0$, the Fritz John con-

ditions become:

$$\begin{aligned} u_0 \nabla f(\mathbf{x}^*) - \sum_{i=1}^m u_i \nabla g_i(\mathbf{x}^*) &= \mathbf{0}, \\ u_i g_i(\mathbf{x}^*) &= 0, \quad \forall i \in \{1, \dots, m\}, \\ u_0, u_i &\geq 0, \quad \forall i \in \{1, \dots, m\}, \\ (u_0, u_1, \dots, u_m) &\neq \mathbf{0}. \end{aligned}$$

The key difference is the sign in the stationarity condition.

15.3.2 Numerical Examples for Fritz John Conditions

Example 15.3. Consider the problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & (x_1 - 3)^2 + (x_2 - 2)^2 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = x_1^2 + x_2^2 - 5 \leq 0, \\ & g_2(\mathbf{x}) = x_1 + 2x_2 - 4 \leq 0, \\ & g_3(\mathbf{x}) = -x_1 \leq 0, \\ & g_4(\mathbf{x}) = -x_2 \leq 0. \end{aligned}$$

Is $\bar{\mathbf{x}} = (2, 1)$ a Fritz John point?

At $\bar{\mathbf{x}} = (2, 1)$:

- $g_1(\bar{\mathbf{x}}) = 4 + 1 - 5 = 0$ (active)
- $g_2(\bar{\mathbf{x}}) = 2 + 2 - 4 = 0$ (active)
- $g_3(\bar{\mathbf{x}}) = -2 < 0$ (inactive)
- $g_4(\bar{\mathbf{x}}) = -1 < 0$ (inactive)

The gradients are:

$$\begin{aligned} \nabla f(\bar{\mathbf{x}}) &= \begin{pmatrix} 2(x_1 - 3) \\ 2(x_2 - 2) \end{pmatrix} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \\ \nabla g_1(\bar{\mathbf{x}}) &= \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}, \\ \nabla g_2(\bar{\mathbf{x}}) &= \begin{pmatrix} 1 \\ 2 \end{pmatrix}. \end{aligned}$$

We need to find $u_0, u_1, u_2 \geq 0$ (with $u_3 = u_4 = 0$ since those constraints are inactive) such that

$$u_0 \begin{pmatrix} -2 \\ -2 \end{pmatrix} + u_1 \begin{pmatrix} 4 \\ 2 \end{pmatrix} + u_2 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

This gives the system:

$$\begin{aligned} -2u_0 + 4u_1 + u_2 &= 0, \\ -2u_0 + 2u_1 + 2u_2 &= 0. \end{aligned}$$

From the second equation: $u_0 = u_1 + u_2$. Substituting into the first: $-2(u_1 + u_2) + 4u_1 + u_2 = 2u_1 - u_2 = 0$, so $u_2 = 2u_1$.

Taking $u_1 = 1$ gives $u_2 = 2$ and $u_0 = 3$. Since all multipliers are nonnegative and not all zero, $\bar{\mathbf{x}} = (2, 1)$ is a Fritz John point.

Is $\bar{\mathbf{x}} = (0, 0)$ a Fritz John point?

At $\bar{\mathbf{x}} = (0, 0)$:

- $g_1(\bar{\mathbf{x}}) = -5 < 0$ (inactive)
- $g_2(\bar{\mathbf{x}}) = -4 < 0$ (inactive)
- $g_3(\bar{\mathbf{x}}) = 0$ (active)
- $g_4(\bar{\mathbf{x}}) = 0$ (active)

The gradients at this point are:

$$\nabla f(\bar{\mathbf{x}}) = \begin{pmatrix} -6 \\ -4 \end{pmatrix}, \quad \nabla g_3(\bar{\mathbf{x}}) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \nabla g_4(\bar{\mathbf{x}}) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}.$$

We need $u_0 \begin{pmatrix} -6 \\ -4 \end{pmatrix} + u_3 \begin{pmatrix} -1 \\ 0 \end{pmatrix} + u_4 \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \mathbf{0}$.

This gives $-6u_0 - u_3 = 0$ and $-4u_0 - u_4 = 0$, so $u_3 = -6u_0$ and $u_4 = -4u_0$. For $u_0 > 0$, we get $u_3, u_4 < 0$, violating nonnegativity. For $u_0 = 0$, we need $u_3 = u_4 = 0$, violating nontriviality.

Therefore, $\bar{\mathbf{x}} = (0, 0)$ is **not** a Fritz John point.

Example 15.4. Consider the problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & -x_1 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = x_2 - (1 - x_1)^3 \leq 0, \\ & g_2(\mathbf{x}) = -x_2 \leq 0. \end{aligned}$$

Is $\bar{\mathbf{x}} = (1, 0)$ a Fritz John point?

At $\bar{\mathbf{x}} = (1, 0)$: Both constraints are active with $g_1(\bar{\mathbf{x}}) = 0$ and $g_2(\bar{\mathbf{x}}) = 0$. The gradients are:

$$\begin{aligned} \nabla f(\bar{\mathbf{x}}) &= \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \\ \nabla g_1(\bar{\mathbf{x}}) &= \begin{pmatrix} 3(1 - x_1)^2 \\ 1 \end{pmatrix} \Big|_{\bar{\mathbf{x}}} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \\ \nabla g_2(\bar{\mathbf{x}}) &= \begin{pmatrix} 0 \\ -1 \end{pmatrix}. \end{aligned}$$

$$\text{We need: } u_0 \begin{pmatrix} -1 \\ 0 \end{pmatrix} + u_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + u_2 \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \mathbf{0}.$$

This gives $-u_0 = 0$ and $u_1 - u_2 = 0$. So $u_0 = 0$ and $u_1 = u_2$. Taking $u_1 = u_2 = 1$ satisfies all conditions.

Therefore, $\bar{\mathbf{x}} = (1, 0)$ is a Fritz John point (with $u_0 = 0$, $u_1 = u_2 = 1$). Note that this is indeed the optimal solution, but the multiplier $u_0 = 0$ means the objective function gradient does not appear in the stationarity condition—a potentially problematic situation.

Example 15.5. Consider the problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & -x_1 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = x_1 + x_2 - 1 \leq 0, \\ & g_2(\mathbf{x}) = -x_2 \leq 0. \end{aligned}$$

Is $\bar{\mathbf{x}} = (1, 0)$ a Fritz John point?

At $\bar{\mathbf{x}} = (1, 0)$: Both constraints are active.

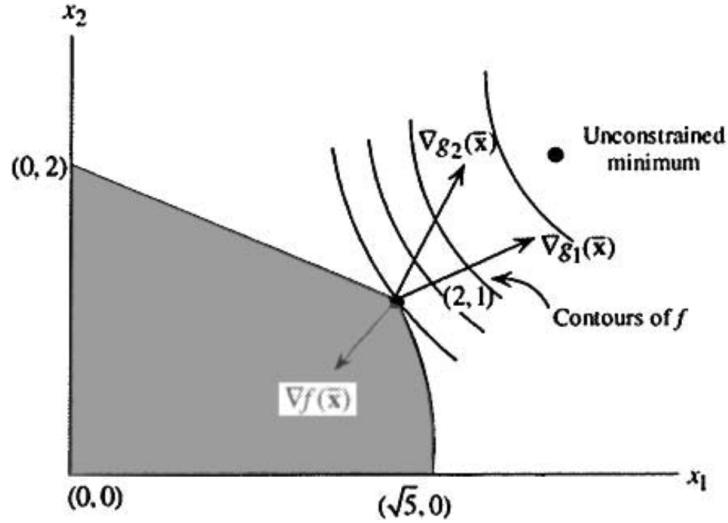


Figure 15.4: Geometric interpretation of the KKT conditions. The negative gradient of the objective function $-\nabla f(\mathbf{x}^*)$ lies in the cone generated by the gradients of the active constraints $\nabla g_i(\mathbf{x}^*)$, $i \in I$.

The gradients are:

$$\nabla f(\bar{\mathbf{x}}) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \nabla g_1(\bar{\mathbf{x}}) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \nabla g_2(\bar{\mathbf{x}}) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}.$$

We need: $u_0 \begin{pmatrix} -1 \\ 0 \end{pmatrix} + u_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + u_2 \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \mathbf{0}$.

This gives $-u_0 + u_1 = 0$ and $u_1 - u_2 = 0$. So $u_0 = u_1 = u_2$. Taking $u_0 = u_1 = u_2 = 1$ satisfies all conditions with $u_0 > 0$.

Therefore, $\bar{\mathbf{x}} = (1, 0)$ is a Fritz John point with $u_0 > 0$.

15.4 Potential Issues with the Fritz John Conditions

The Fritz John conditions have some undesirable properties that limit their usefulness as optimality conditions.

15.4.1 Trivial Satisfaction of FJ Conditions

Points can satisfy the FJ conditions **trivially** in the following cases:

1. **Zero gradient of objective or constraint:** If a feasible point \mathbf{x}^* satisfies $\nabla f(\mathbf{x}^*) = \mathbf{0}$ or $\nabla g_i(\mathbf{x}^*) = \mathbf{0}$ for some $i \in I$, the FJ conditions can be satisfied by setting $u_0 > 0$ (in the first case) or $u_i > 0$ (in the second case) and all other multipliers to zero.
2. **Replacing equality by inequality constraints:** If an equality constraint $g(\mathbf{x}) = 0$ is replaced by the pair $g_1(\mathbf{x}) = g(\mathbf{x}) \leq 0$ and $g_2(\mathbf{x}) = -g(\mathbf{x}) \leq 0$, then the FJ conditions are trivially satisfied at *all* feasible points by setting $u_1 = u_2 = \alpha > 0$ and all other multipliers to zero.
3. **Adding redundant constraints:** Given *any* feasible solution $\bar{\mathbf{x}}$, adding the trivial redundant constraint $g(\mathbf{x}) \equiv -\|\mathbf{x} - \bar{\mathbf{x}}\|^2 \leq 0$ (which is binding at $\bar{\mathbf{x}}$ with gradient $\nabla g(\bar{\mathbf{x}}) = \mathbf{0}$) makes $\bar{\mathbf{x}}$ a FJ point.

15.4.2 FJ Points Can Be Nonoptimal for Linear Programs

A particularly troubling issue is that Fritz John points may not be optimal even for linear programs.

Example 15.6 (Nonoptimal FJ Point for LP). Consider a linear program where a vertex \mathbf{x}^* is a FJ point but not optimal. This can occur when $u_0 = 0$ in the FJ conditions, meaning the objective function plays no role in the stationarity condition. The point satisfies the FJ conditions purely based on the geometry of the constraints.

This motivates the need for stronger conditions that ensure $u_0 > 0$.

15.5 Karush-Kuhn-Tucker (KKT) Necessary Conditions

The Karush-Kuhn-Tucker (KKT) conditions strengthen the Fritz John conditions by requiring $u_0 > 0$. This ensures that the objective function gradient plays a meaningful role in the optimality conditions.

15.5.1 Statement of the KKT Conditions

Theorem 15.5 (KKT Necessary Conditions). *Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, consider the constrained problem*

$$\min\{f(\mathbf{x}) : g_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}, \mathbf{x} \in X\}.$$

Suppose the gradients $\nabla g_i(\mathbf{x}^)$, $i \in I$, are **linearly independent** at a feasible point \mathbf{x}^* . If \mathbf{x}^* is a local minimum, then there exist scalars u_i , $i \in \{1, \dots, m\}$, such that*

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^m u_i \nabla g_i(\mathbf{x}^*) = \mathbf{0}, \quad (15.5)$$

$$u_i g_i(\mathbf{x}^*) = 0, \quad \forall i \in \{1, \dots, m\}, \quad (15.6)$$

$$u_i \geq 0, \quad \forall i \in \{1, \dots, m\}. \quad (15.7)$$

The key differences from the Fritz John conditions are:

1. There is no multiplier u_0 for the objective function—equivalently, we have $u_0 = 1$.
2. The nontriviality condition $(u_0, u_1, \dots, u_m) \neq \mathbf{0}$ is automatically satisfied.
3. We require a **constraint qualification**: the gradients of active constraints must be linearly independent.

Definition 15.6 (Linear Independence Constraint Qualification (LICQ)). The **Linear Independence Constraint Qualification (LICQ)** holds at a feasible point \mathbf{x}^* if the gradients $\nabla g_i(\mathbf{x}^*)$, $i \in I$, of the active constraints are linearly independent.

LICQ is analogous to the regularity condition in the necessary optimality conditions for equality-constrained problems.

Remark 15.6. The KKT conditions were derived independently by William Karush in his 1939 master's thesis and by Harold Kuhn and Albert Tucker in 1951.

15.5.2 Terminology for KKT Conditions

Definition 15.7 (KKT Optimality Conditions). The conditions in Theorem 15.5 consist of:

- **Lagrange Multipliers:** The scalars $u_i, i \in \{1, \dots, m\}$.
- **Primal Feasibility:** $\mathbf{x}^* \in X$ and $g_i(\mathbf{x}^*) \leq 0$ for all $i \in \{1, \dots, m\}$.
- **Stationarity (Dual Feasibility):** $\nabla f(\mathbf{x}^*) + \sum_{i=1}^m u_i \nabla g_i(\mathbf{x}^*) = 0$.
- **Dual Nonnegativity:** $u_i \geq 0$ for all $i \in \{1, \dots, m\}$.
- **Complementary Slackness:** $u_i g_i(\mathbf{x}^*) = 0$ for each i .

Together, these conditions are called the **Karush-Kuhn-Tucker (KKT) Optimality Conditions**.

Definition 15.8 (KKT Point). A feasible point \mathbf{x}^* is called a **KKT point** if there exist Lagrange multipliers (u_1^*, \dots, u_m^*) such that $(\mathbf{x}^*, u_1^*, \dots, u_m^*)$ satisfies the KKT conditions.

15.5.3 Numerical Examples for KKT Conditions

Example 15.7. Returning to Example 15.3, we check whether the points are KKT points.

At $\bar{\mathbf{x}} = (2, 1)$: We found earlier that the FJ conditions are satisfied with $u_0 = 3, u_1 = 1, u_2 = 2, u_3 = u_4 = 0$. Since $u_0 > 0$, we can normalize by dividing all multipliers by $u_0 = 3$ to get the KKT multipliers: $u_1 = 1/3, u_2 = 2/3$.

Alternatively, verify directly: $\nabla f(\bar{\mathbf{x}}) + u_1 \nabla g_1(\bar{\mathbf{x}}) + u_2 \nabla g_2(\bar{\mathbf{x}}) = \begin{pmatrix} -2 \\ -2 \end{pmatrix} + \frac{1}{3} \begin{pmatrix} 4 \\ 2 \end{pmatrix} + \frac{2}{3} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Therefore, $\bar{\mathbf{x}} = (2, 1)$ is a KKT point.

At $\bar{\mathbf{x}} = (0, 0)$: We showed this is not a FJ point, so it cannot be a KKT point.

Example 15.8. Returning to Example 15.4, we found that $\bar{\mathbf{x}} = (1, 0)$ is a FJ point with $u_0 = 0$. Since $u_0 = 0$, we cannot normalize to obtain KKT multipliers.

Moreover, LICQ fails at this point because $\nabla g_1(\bar{\mathbf{x}}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $\nabla g_2(\bar{\mathbf{x}}) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ are linearly dependent (they are scalar multiples of each other).

The KKT conditions require: $\begin{pmatrix} -1 \\ 0 \end{pmatrix} + u_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + u_2 \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \mathbf{0}$.

This gives $-1 = 0$ (first component), which is impossible. Therefore, $\bar{\mathbf{x}} = (1, 0)$ is **not** a KKT point, even though it is the optimal solution! This example shows that the KKT conditions may fail to identify an optimal solution when LICQ does not hold.

Example 15.9. Returning to Example 15.5, we found $\bar{\mathbf{x}} = (1, 0)$ is a FJ point with $u_0 = u_1 = u_2 = 1$. Since $u_0 = 1 > 0$, the KKT multipliers are $u_1 = u_2 = 1$.

Verify LICQ: $\nabla g_1(\bar{\mathbf{x}}) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\nabla g_2(\bar{\mathbf{x}}) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ are linearly independent.

Therefore, $\bar{\mathbf{x}} = (1, 0)$ is a KKT point.

15.6 KKT Conditions for Linear Programs

For linear programs, the KKT conditions take a particularly simple form and are both necessary and sufficient for optimality.

Consider the standard form linear program

$$\min\{\mathbf{c}^T \mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\},$$

where $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$.

Writing this with inequality constraints $g_i(\mathbf{x}) = -x_i \leq 0$ for $i \in \{1, \dots, n\}$ and equality constraints $A\mathbf{x} = \mathbf{b}$, the KKT conditions become:

- **Primal Feasibility:** $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.
- **Stationarity:** $\mathbf{c} - A^T \boldsymbol{\lambda} - \boldsymbol{\mu} = \mathbf{0}$, or equivalently, $\mathbf{c} = A^T \boldsymbol{\lambda} + \boldsymbol{\mu}$, where $\boldsymbol{\lambda} \in \mathbb{R}^m$ are the multipliers for the equality constraints and $\boldsymbol{\mu} \in \mathbb{R}^n$ are the multipliers for the nonnegativity constraints.

- **Dual Nonnegativity:** $\mu \geq 0$.
- **Complementary Slackness:** $\mu_i x_i = 0$ for each $i \in \{1, \dots, n\}$.

The dual LP is $\max\{\mathbf{b}^T \boldsymbol{\lambda} : A^T \boldsymbol{\lambda} \leq \mathbf{c}\}$, and if we let $\boldsymbol{\mu} = \mathbf{c} - A^T \boldsymbol{\lambda}$ be the dual slack variables, the KKT conditions are exactly the conditions for primal-dual optimality in linear programming.

Theorem 15.6. *For linear programs, the KKT conditions are both necessary and sufficient for optimality. That is, a feasible point \mathbf{x}^* is optimal if and only if it is a KKT point.*

15.7 Geometric Interpretation of the KKT Conditions

The KKT stationarity condition $\nabla f(\mathbf{x}^*) + \sum_{i \in I} u_i \nabla g_i(\mathbf{x}^*) = \mathbf{0}$ has an important geometric interpretation.

Remark 15.7 (Geometric Interpretation). Rearranging the stationarity condition gives

$$-\nabla f(\mathbf{x}^*) = \sum_{i \in I} u_i \nabla g_i(\mathbf{x}^*).$$

Since $u_i \geq 0$ for all $i \in I$, this says that $-\nabla f(\mathbf{x}^*)$ lies in the **cone spanned by the gradients of the active constraints**:

$$-\nabla f(\mathbf{x}^*) \in \text{cone}\{\nabla g_i(\mathbf{x}^*) : i \in I\}.$$

Geometrically, the gradient $\nabla g_i(\mathbf{x}^*)$ points outward from the feasible region (in the direction of increasing g_i). The KKT conditions require that the negative gradient of the objective function (the direction of steepest descent) can be written as a nonnegative combination of these outward-pointing constraint gradients.

This means that at a KKT point, the direction of improvement for the objective function points “into” the constraint boundaries, so no feasible descent direction exists.

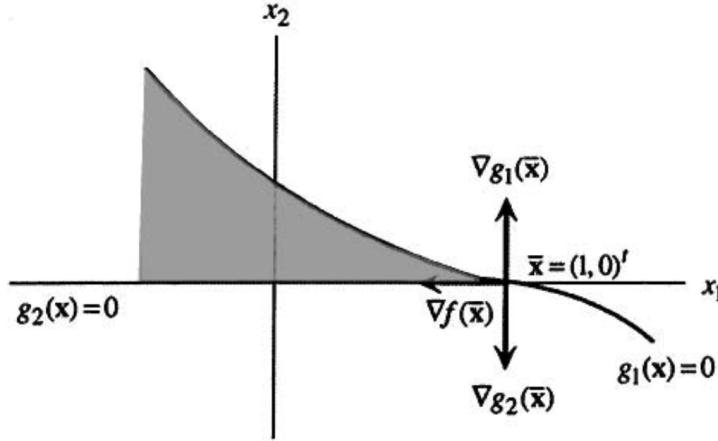


Figure 15.5: Visualization of the KKT stationarity condition. The gradient $\nabla f(\mathbf{x}^*)$ is expressed as a nonnegative combination of the constraint gradients, meaning the descent direction is blocked by the active constraints.

15.7.1 Insights into the KKT Conditions

Theorem 15.7 (Characterization of KKT Points). *Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, let \mathbf{x}^* be a feasible solution to*

$$\min\{f(\mathbf{x}) : g_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}, \mathbf{x} \in X\}.$$

Define

$$\begin{aligned} F_0 &= \{\mathbf{d} \in \mathbb{R}^n : \nabla f(\mathbf{x}^*)^T \mathbf{d} < 0\}, \\ G'_0 &= \{\mathbf{d} \in \mathbb{R}^n : \mathbf{d} \neq \mathbf{0}, \nabla g_i(\mathbf{x}^*)^T \mathbf{d} \leq 0, i \in I\}. \end{aligned}$$

Then

$$\mathbf{x}^* \text{ is a KKT point} \iff F_0 \cap G'_0 = \emptyset.$$

Moreover, \mathbf{x}^* is a KKT point if and only if it solves the following linear program:

$$\min\{f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) : g_i(\mathbf{x}^*) + \nabla g_i(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) \leq 0, i \in \{1, \dots, m\}\}.$$

The linear program in this theorem is the first-order (linear) approximation to the original nonlinear program at \mathbf{x}^* . This provides a computational

interpretation: a point is a KKT point if and only if it cannot be improved by solving the linearized problem.

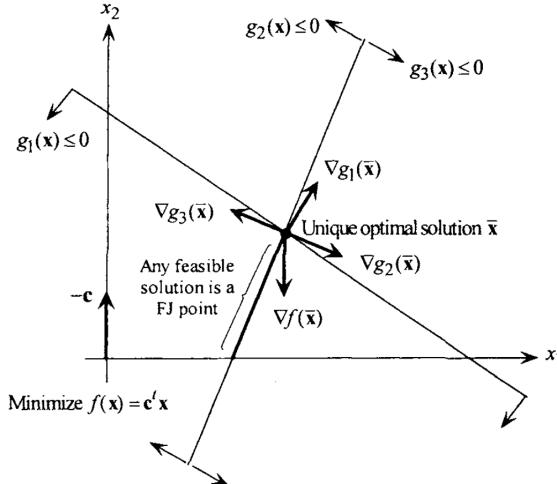


Figure 4.9 FJ conditions are not sufficient for optimality for LP problems.

Figure 15.6: Example illustrating the KKT conditions for a problem with multiple active constraints. The optimal point \mathbf{x}^* satisfies the stationarity condition where $-\nabla f(\mathbf{x}^*)$ is a conic combination of the active constraint gradients.

15.8 Constraint Qualifications

The KKT conditions are necessary for optimality only when an appropriate constraint qualification holds. LICQ is one such condition, but there are others that can be useful when LICQ fails.

Definition 15.9 (Constraint Qualification). A **constraint qualification** is a condition on the constraints at a feasible point that ensures the KKT conditions are necessary for local optimality.

15.8.1 Alternative Constraint Qualifications

The following constraint qualifications do **not** require the gradients of active constraints to be linearly independent.

Theorem 15.8. *The KKT conditions are necessary for local minima under any of the following constraint qualifications. Assume $X \subseteq \mathbb{R}^n$ is nonempty and open, and the functions f and g_i , $i \in \{1, \dots, m\}$, are continuously differentiable.*

1. **Linearly-Constrained Problems:** All constraint functions g_i , $i \in \{1, \dots, m\}$, are **affine** (linear plus constant).
2. **Slater's Condition:** The functions g_i , $i \in \{1, \dots, m\}$, are **convex** and there exists $\bar{\mathbf{x}} \in X$ such that $g_i(\bar{\mathbf{x}}) < 0$ for each $i \in \{1, \dots, m\}$. (Such a point $\bar{\mathbf{x}}$ is called a **strictly feasible point** or **Slater point**.)
3. **Mangasarian-Fromovitz Constraint Qualification (MFCQ):** There exists $\mathbf{d} \in \mathbb{R}^n$ such that $\nabla g_i(\mathbf{x}^*)^T \mathbf{d} < 0$ for each $i \in I$.

Remark 15.8. All of these constraint qualifications can be generalized to problems with both equality and inequality constraints.

The relationships between these constraint qualifications are:

- LICQ implies MFCQ.
- For convex problems, Slater's condition implies MFCQ.
- Affine constraints automatically satisfy MFCQ (and hence LICQ is not needed).

15.9 KKT Conditions for Convex Optimization

For convex optimization problems, the KKT conditions have special significance: they are sufficient for global optimality (under mild conditions), though they may not always be necessary.

15.9.1 KKT Conditions Are Not Always Necessary for Convex Problems

Even for convex optimization problems, the KKT conditions may fail to hold at an optimal solution if no constraint qualification is satisfied.

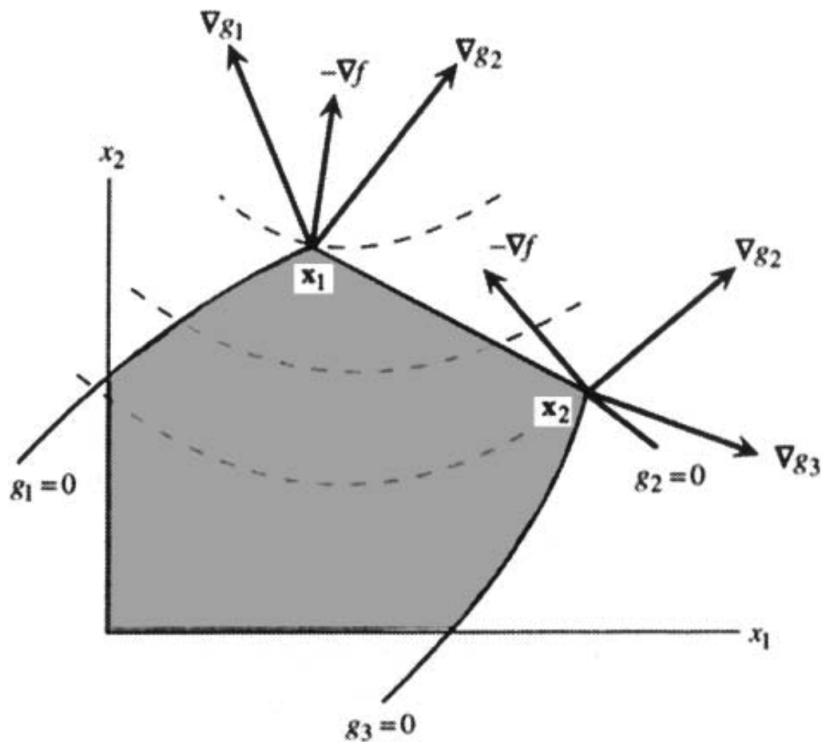


Figure 15.7: Illustration of different constraint qualifications. When LICQ holds, the active constraint gradients are linearly independent, ensuring the KKT conditions are necessary for local optimality.

Example 15.10 (KKT Conditions Not Necessary). Consider the convex optimization problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & x_1 \\ \text{s.t.} \quad & (x_1 - 1)^2 + (x_2 - 1)^2 \leq 1, \\ & (x_1 - 1)^2 + (x_2 + 1)^2 \leq 1. \end{aligned}$$

The feasible region is the intersection of two disks, both centered at $x_1 = 1$. The intersection points are $(0, 0)$ and $(2, 0)$. The optimal solution is $\mathbf{x}^* = (0, 0)$ (minimizing x_1).

At $\mathbf{x}^* = (0, 0)$:

- Both constraints are active.
- $\nabla g_1(\mathbf{x}^*) = 2 \begin{pmatrix} 0 - 1 \\ 0 - 1 \end{pmatrix} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$
- $\nabla g_2(\mathbf{x}^*) = 2 \begin{pmatrix} 0 - 1 \\ 0 + 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \end{pmatrix}$

These gradients are linearly independent, so LICQ holds. The KKT conditions require:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} + u_1 \begin{pmatrix} -2 \\ -2 \end{pmatrix} + u_2 \begin{pmatrix} -2 \\ 2 \end{pmatrix} = \mathbf{0}.$$

This gives: $1 - 2u_1 - 2u_2 = 0$ and $-2u_1 + 2u_2 = 0$. From the second equation, $u_1 = u_2$. Substituting: $1 - 4u_1 = 0$, so $u_1 = u_2 = 1/4 > 0$.

In this case, the KKT conditions **are** satisfied. However, for problems where constraint qualifications fail, the KKT conditions may not hold at the optimum.

15.9.2 KKT Conditions Are Sufficient for Convex Problems

The most important property of the KKT conditions for convex optimization is that they are sufficient for global optimality.

Theorem 15.9 (KKT Sufficient Conditions for Convex Optimization). *Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and*

$g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, consider the constrained problem

$$\min\{f(\mathbf{x}) : g_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}, \mathbf{x} \in X\}.$$

Let I denote the indices of the active constraints at a feasible point \mathbf{x}^* . Suppose the functions f and g_i , $i \in I$, are convex. If \mathbf{x}^* is a KKT point, then \mathbf{x}^* is a **global minimum** of the problem.

Proof. Let \mathbf{x}^* be a KKT point with multipliers $u_i \geq 0$. Define the Lagrangian function

$$L(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^m u_i g_i(\mathbf{x}).$$

Since f is convex and $u_i \geq 0$ with g_i convex for $i \in I$ (and $u_i = 0$ for $i \notin I$ by complementary slackness), the Lagrangian L is a convex function.

The stationarity condition $\nabla f(\mathbf{x}^*) + \sum_{i=1}^m u_i \nabla g_i(\mathbf{x}^*) = \mathbf{0}$ means $\nabla L(\mathbf{x}^*) = \mathbf{0}$. Since L is convex, this implies \mathbf{x}^* is a global minimizer of L .

For any feasible \mathbf{x} :

$$f(\mathbf{x}) \geq f(\mathbf{x}) + \sum_{i=1}^m u_i g_i(\mathbf{x}) = L(\mathbf{x}) \geq L(\mathbf{x}^*) = f(\mathbf{x}^*) + \sum_{i=1}^m u_i g_i(\mathbf{x}^*) = f(\mathbf{x}^*),$$

where the first inequality uses $u_i \geq 0$ and $g_i(\mathbf{x}) \leq 0$, and the last equality uses complementary slackness.

Therefore, \mathbf{x}^* is a global minimum. \square

Remark 15.9. For convex optimization problems where a constraint qualification (such as Slater's condition) holds, the KKT conditions are both necessary and sufficient for global optimality. This makes finding KKT points equivalent to solving the optimization problem.

15.10 Summary

This chapter developed the theory of optimality conditions for inequality-constrained optimization problems:

1. **Geometric Necessary Conditions:** At a local minimum, the cone of improving directions F_0 and the cone of feasible directions D must have empty intersection.

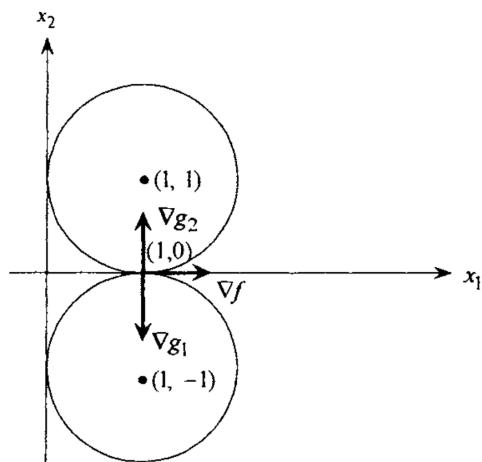


Figure 4.11 KKT conditions are not necessary for convex programming problems.

Figure 15.8: KKT conditions for convex optimization. For convex problems satisfying a constraint qualification, any KKT point is a global minimum. The figure shows the feasible region, level curves of the objective, and the optimal point.

2. **Fritz John Conditions:** These algebraic conditions involve multipliers $u_0, u_1, \dots, u_m \geq 0$ (not all zero) and are necessary for local optimality. However, they can be satisfied trivially and may identify nonoptimal points.
3. **KKT Conditions:** By requiring $u_0 = 1$ (or equivalently, $u_0 > 0$), the KKT conditions ensure that the objective function gradient plays a role in the optimality conditions. They require a constraint qualification such as LICQ to be necessary.
4. **Constraint Qualifications:** LICQ (linear independence of active constraint gradients), Slater's condition (existence of strictly feasible point for convex constraints), and MFCQ are conditions that ensure the KKT conditions are necessary for local optimality.
5. **Convex Optimization:** For convex problems, the KKT conditions are sufficient for global optimality. Combined with a constraint qualification, they become both necessary and sufficient.
6. **Linear Programming:** The KKT conditions are both necessary and sufficient for optimality of linear programs, providing the theoretical foundation for LP algorithms.

The KKT conditions form the cornerstone of nonlinear programming theory and are fundamental to the development of algorithms for constrained optimization.

Chapter 16

KKT Conditions: Mixed Constraints and Summary

This chapter extends the Karush-Kuhn-Tucker (KKT) theory to optimization problems with both inequality and equality constraints. We develop geometric necessary conditions, state the KKT necessary conditions for mixed constraints, and present sufficient conditions for convex problems. We then establish second-order necessary and sufficient conditions for general nonlinear constrained optimization. The chapter concludes with numerical examples and a synthesis of the key concepts from the course.

Recommended Reading

- Sections 4.3 and 4.4 of Bazaraa, Sherali, and Shetty (2006)
- Chapter 12 of Nocedal and Wright

16.1 Review: KKT Necessary Conditions for Inequality Constraints

Before extending to mixed constraints, we recall the KKT conditions for problems with only inequality constraints.

Theorem 16.1 (KKT Necessary Conditions for Inequality-Constrained Problems). *Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, consider the constrained*

problem

$$\min\{f(\mathbf{x}) : g_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}, \mathbf{x} \in X\}.$$

Suppose $\nabla g_i(\mathbf{x}^*)$, $i \in I$, are linearly independent, where $I = \{i : g_i(\mathbf{x}^*) = 0\}$ is the set of active constraint indices. If \mathbf{x}^* is a local minimum, then there exist scalars u_i , $i \in \{1, \dots, m\}$, such that

$$\begin{aligned}\nabla f(\mathbf{x}^*) + \sum_{i=1}^m u_i \nabla g_i(\mathbf{x}^*) &= \mathbf{0}, \\ u_i g_i(\mathbf{x}^*) &= 0, \quad \forall i \in \{1, \dots, m\}, \\ u_i &\geq 0, \quad \forall i \in \{1, \dots, m\}.\end{aligned}$$

Definition 16.1 (KKT Point). Any point \mathbf{x}^* for which there exist Lagrange multipliers (u_1^*, \dots, u_m^*) such that $(\mathbf{x}^*, u_1^*, \dots, u_m^*)$ satisfy the KKT conditions is called a **KKT point**.

16.2 Problems with Both Inequality and Equality Constraints

We now consider the more general constrained optimization problem that includes both inequality and equality constraints:

$$\begin{aligned}\min_{\mathbf{x} \in X} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad \forall i \in \{1, \dots, m\}, \\ & h_j(\mathbf{x}) = 0, \quad \forall j \in \{1, \dots, l\}.\end{aligned}\tag{16.1}$$

16.2.1 Key Sets for Mixed Constraints

At a feasible point \mathbf{x}^* , we define the following sets that characterize directions of potential improvement:

Definition 16.2 (Descent and Feasible Direction Sets). Let \mathbf{x}^* be a feasible point for problem (16.1). Define:

- $F_0 := \{\mathbf{d} \in \mathbb{R}^n : \nabla f(\mathbf{x}^*)^T \mathbf{d} < 0\}$ — the set of descent directions for the objective.

- $G_0 := \{\mathbf{d} \in \mathbb{R}^n : \nabla g_i(\mathbf{x}^*)^T \mathbf{d} < 0 \text{ for each } i \in I\}$, where $I = \{i \in \{1, \dots, m\} : g_i(\mathbf{x}^*) = 0\}$ — the set of directions improving all active inequality constraints.
- $H_0 := \{\mathbf{d} \in \mathbb{R}^n : \nabla h_j(\mathbf{x}^*)^T \mathbf{d} = 0 \text{ for each } j \in \{1, \dots, l\}\}$ — the set of directions preserving the equality constraints (to first order).

Remark 16.1 (Differentiability Assumptions). Throughout this chapter, we assume that f and g_i , $i \in \{1, \dots, m\}$, are differentiable, and that h_j , $j \in \{1, \dots, l\}$, are continuously differentiable.

16.3 Geometric Necessary Optimality Condition

The geometric necessary condition provides intuition for why certain points cannot be local minima.

Theorem 16.2 (Geometric Necessary Condition). *Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, and $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j \in \{1, \dots, l\}$, consider the constrained problem (16.1).*

Suppose $\nabla h_j(\mathbf{x}^)$, $j \in \{1, \dots, l\}$, are linearly independent. If \mathbf{x}^* is a local minimum, then*

$$F_0 \cap G_0 \cap H_0 = \emptyset.$$

The geometric interpretation is intuitive: if \mathbf{x}^* is a local minimum, there cannot exist a direction that simultaneously decreases the objective function, improves all active inequality constraints, and preserves the equality constraints. If such a direction existed, we could move in that direction to find a nearby feasible point with lower objective value.

Remark 16.2. We proceed directly to the KKT necessary conditions for problems with mixed constraints, skipping the Fritz John conditions that would serve as an intermediate step.

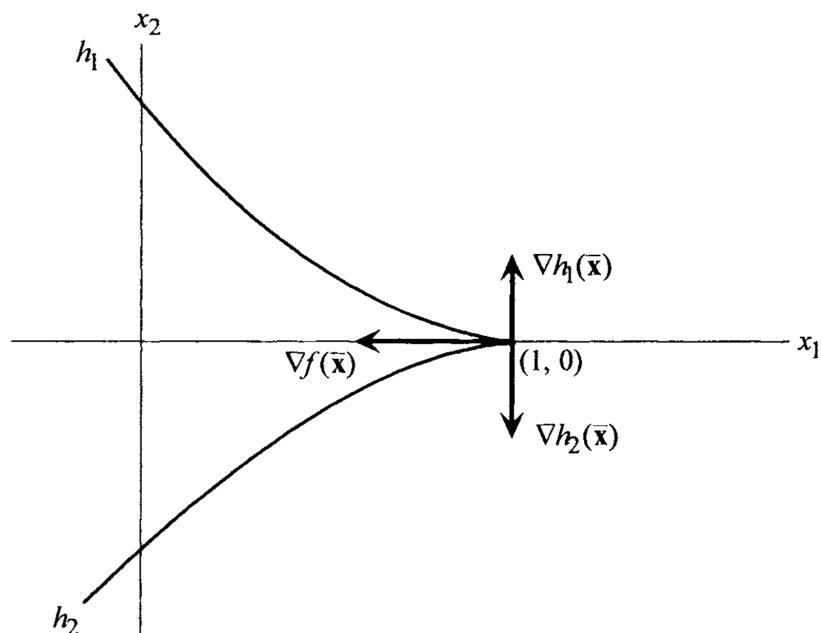


Figure 16.1: Geometric necessary condition for problems with mixed constraints. At a local minimum \mathbf{x}^* , the intersection $F_0 \cap G_0 \cap H_0 = \emptyset$, meaning no direction can simultaneously decrease the objective, satisfy the active inequality constraints, and preserve the equality constraints.

16.4 KKT Necessary Conditions for Mixed Constraints

Theorem 16.3 (KKT Necessary Conditions for Mixed Constraints).

Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, and $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j \in \{1, \dots, l\}$, consider the constrained problem

$$\min_{\mathbf{x} \in X} \left\{ f(\mathbf{x}) : g_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}, h_j(\mathbf{x}) = 0, j \in \{1, \dots, l\} \right\}.$$

Suppose $\nabla g_i(\mathbf{x}^*)$, $i \in I$, and $\nabla h_j(\mathbf{x}^*)$, $j \in \{1, \dots, l\}$, are (jointly) linearly independent, where I is the set of active inequality constraint indices. If \mathbf{x}^* is a local minimum, then there exist scalars u_i , $i \in \{1, \dots, m\}$, and v_j , $j \in \{1, \dots, l\}$, such that

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^m u_i \nabla g_i(\mathbf{x}^*) + \sum_{j=1}^l v_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}, \quad (16.2)$$

$$u_i g_i(\mathbf{x}^*) = 0, \quad \forall i \in \{1, \dots, m\}, \quad (16.3)$$

$$u_i \geq 0, \quad \forall i \in \{1, \dots, m\}. \quad (16.4)$$

16.4.1 Understanding the KKT Conditions for Mixed Constraints

Several important observations help in understanding and applying these conditions:

1. **Complementary slackness is redundant for equality constraints.** Since $h_j(\mathbf{x}^*) = 0$ for all feasible points, a complementary slackness condition $v_j h_j(\mathbf{x}^*) = 0$ would be automatically satisfied for any value of v_j .
2. **The multipliers for equality constraints are unrestricted in sign.** Unlike the multipliers $u_i \geq 0$ for inequality constraints, the multipliers v_j can take any real value.
3. **Geometric interpretation:** The stationarity condition (16.2) states that $-\nabla f(\mathbf{x}^*)$ is a conic combination of the gradients of the binding

inequality constraints at \mathbf{x}^* , plus a linear combination of the gradients of the equality constraints at \mathbf{x}^* .

4. **Special cases:** Ignoring either the equality constraints (setting $l = 0$) or the inequality constraints (setting $m = 0$) yields the KKT conditions we encountered earlier for those special cases.
5. **Derivation from inequality-only case:** The KKT conditions for mixed constraints can be derived from the inequality-only case by replacing each equality constraint $h_j(\mathbf{x}) = 0$ with two inequality constraints $h_j(\mathbf{x}) \leq 0$ and $-h_j(\mathbf{x}) \leq 0$. The corresponding multipliers are then combined as $v_j = u_j^+ - u_j^-$, where $u_j^+, u_j^- \geq 0$ are the multipliers for the two inequality constraints.

Remark 16.3 (KKT Conditions for Maximization). For maximization problems, the KKT conditions change in the following ways:

- The stationarity condition becomes: $-\nabla f(\mathbf{x}^*) + \sum_{i=1}^m u_i \nabla g_i(\mathbf{x}^*) + \sum_{j=1}^l v_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$.
- Alternatively, one can convert the maximization to minimization by negating the objective.

16.5 Numerical Examples

Example 16.1 (Verifying a KKT Point). Consider the optimization problem:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & (x_1 - 3)^2 + (x_2 - 2)^2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 \leq 5 \\ & -x_1 \leq 0 \\ & -x_2 \leq 0 \\ & x_1 + 2x_2 = 4. \end{aligned}$$

We verify whether $\mathbf{x}^* = (2, 1)^T$ is a KKT point.

Step 1: Check feasibility.

- $g_1(\mathbf{x}^*) = 4 + 1 - 5 = 0$ (active)
- $g_2(\mathbf{x}^*) = -2 < 0$ (inactive)

- $g_3(\mathbf{x}^*) = -1 < 0$ (inactive)
- $h_1(\mathbf{x}^*) = 2 + 2 - 4 = 0$ (satisfied)

The point is feasible with $I = \{1\}$.

Step 2: Compute gradients.

$$\begin{aligned}\nabla f(\mathbf{x}^*) &= \begin{pmatrix} 2(x_1 - 3) \\ 2(x_2 - 2) \end{pmatrix} \Big|_{\mathbf{x}^*} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \\ \nabla g_1(\mathbf{x}^*) &= \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} \Big|_{\mathbf{x}^*} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}, \\ \nabla g_2(\mathbf{x}^*) &= \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \nabla g_3(\mathbf{x}^*) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \\ \nabla h_1(\mathbf{x}^*) &= \begin{pmatrix} 1 \\ 2 \end{pmatrix}.\end{aligned}$$

Step 3: Check constraint qualification. The active constraint gradients are $\nabla g_1(\mathbf{x}^*) = (4, 2)^T$ and $\nabla h_1(\mathbf{x}^*) = (1, 2)^T$. These are linearly independent (not parallel), so the constraint qualification is satisfied.

Step 4: Solve the KKT stationarity condition. We need $u_1, u_2, u_3 \geq 0$ and $v_1 \in \mathbb{R}$ such that:

$$\begin{pmatrix} -2 \\ -2 \end{pmatrix} + u_1 \begin{pmatrix} 4 \\ 2 \end{pmatrix} + u_2 \begin{pmatrix} -1 \\ 0 \end{pmatrix} + u_3 \begin{pmatrix} 0 \\ -1 \end{pmatrix} + v_1 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

By complementary slackness, $u_2 = u_3 = 0$ (since g_2 and g_3 are inactive). The system reduces to:

$$\begin{aligned}-2 + 4u_1 + v_1 &= 0, \\ -2 + 2u_1 + 2v_1 &= 0.\end{aligned}$$

From the second equation: $u_1 + v_1 = 1$. Substituting into the first: $-2 + 4u_1 + (1 - u_1) = 0$, giving $3u_1 = 1$, so $u_1 = 1/3$ and $v_1 = 2/3$. Since $u_1 = 1/3 > 0$, all KKT conditions are satisfied. Therefore, $\mathbf{x}^* = (2, 1)^T$ is a KKT point with multipliers $u^* = (1/3, 0, 0)^T$ and $v^* = 2/3$.

Example 16.2 (Failure of Constraint Qualification). Consider the problem:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & -x_1 \\ \text{s.t.} \quad & x_2 - (1 - x_1)^3 = 0 \\ & -x_2 - (1 - x_1)^3 = 0. \end{aligned}$$

We investigate whether $\mathbf{x}^* = (1, 0)^T$ is a KKT point.

Step 1: Check feasibility. Both constraints give $0 - 0 = 0$, so \mathbf{x}^* is feasible.

Step 2: Compute gradients.

$$\begin{aligned} \nabla f(\mathbf{x}^*) &= \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \\ \nabla h_1(\mathbf{x}^*) &= \begin{pmatrix} 3(1 - x_1)^2 \\ 1 \end{pmatrix} \Big|_{\mathbf{x}^*} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \\ \nabla h_2(\mathbf{x}^*) &= \begin{pmatrix} 3(1 - x_1)^2 \\ -1 \end{pmatrix} \Big|_{\mathbf{x}^*} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}. \end{aligned}$$

Step 3: Check constraint qualification. The gradients $\nabla h_1(\mathbf{x}^*) = (0, 1)^T$ and $\nabla h_2(\mathbf{x}^*) = (0, -1)^T$ are linearly dependent (one is a scalar multiple of the other). The constraint qualification fails.

Step 4: Attempt to satisfy KKT conditions. The stationarity condition requires:

$$\begin{pmatrix} -1 \\ 0 \end{pmatrix} + v_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + v_2 \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

The first component gives $-1 = 0$, which is impossible. Therefore, no multipliers exist that satisfy the KKT conditions.

This example demonstrates that when the constraint qualification fails, a local minimum may not satisfy the KKT necessary conditions. The point $\mathbf{x}^* = (1, 0)^T$ is actually the global minimum of this problem, but it is not a KKT point because the constraint qualification is violated.

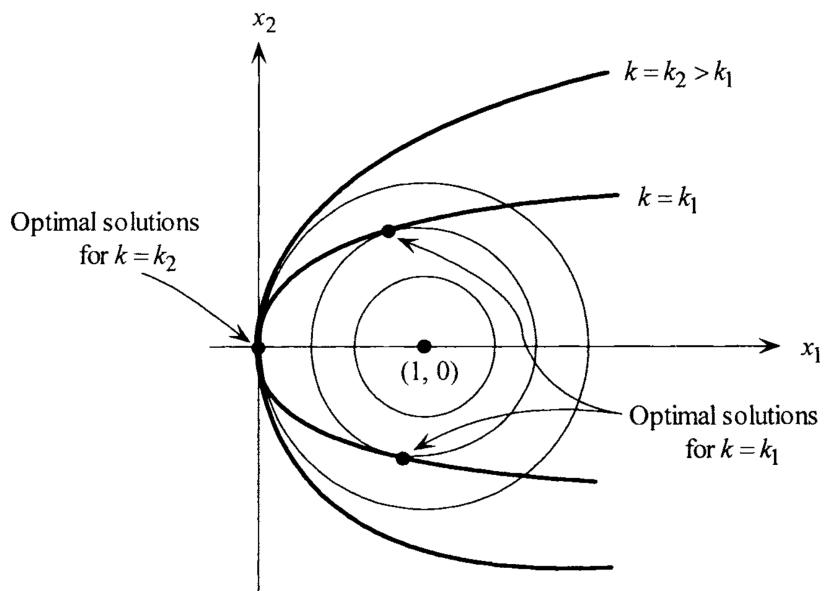


Figure 16.2: KKT conditions for mixed constraints: numerical example. The figure illustrates the feasible region defined by both inequality and equality constraints, showing the optimal point and the gradients of the active constraints.

16.6 KKT Sufficient Conditions for Convex Problems

While the KKT conditions are necessary for local optimality under appropriate constraint qualifications, they become sufficient under convexity assumptions.

Theorem 16.4 (KKT Sufficient Conditions). *Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, and $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j \in \{1, \dots, l\}$, consider the constrained problem*

$$\min_{\mathbf{x} \in X} \left\{ f(\mathbf{x}) : g_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}, h_j(\mathbf{x}) = 0, j \in \{1, \dots, l\} \right\}.$$

Let I denote the indices of the active inequality constraints at the feasible point \mathbf{x}^ . Suppose the functions f and g_i , $i \in I$, are convex and the functions h_j , $j \in \{1, \dots, l\}$, are affine. If \mathbf{x}^* is a KKT point, then it is a global minimum of the above problem.*

Remark 16.4 (KKT Conditions and Convex Optimization). Two important cautions regarding KKT conditions and convex optimization:

1. **KKT conditions are sufficient for convex problems:** When the objective and active inequality constraints are convex and the equality constraints are affine, any KKT point is a global minimum.
2. **KKT conditions are not always necessary for convex problems:** The KKT conditions require a constraint qualification (such as linear independence of active constraint gradients). Even for convex problems, if the constraint qualification fails, a global minimum may not be a KKT point.

16.7 KKT Second-Order Conditions

We now develop second-order conditions that provide sharper characterizations of local optimality. Consider the problem (16.1) and assume that f , $\mathbf{g} = (g_1, \dots, g_m)^T$, and $\mathbf{h} = (h_1, \dots, h_l)^T$ are twice continuously differentiable.

16.7.1 The Restricted Lagrangian and Critical Cone

Definition 16.3 (Restricted Lagrangian). Suppose \mathbf{x}^* is a KKT point with Lagrange multipliers $\mathbf{u}^* \in \mathbb{R}^m$ and $\mathbf{v}^* \in \mathbb{R}^l$ corresponding to the inequality and equality constraints, respectively. Let $I \subseteq \{1, \dots, m\}$ denote the set of active inequality constraint indices at \mathbf{x}^* .

The **restricted Lagrangian** is defined as:

$$L(\mathbf{x}, \mathbf{u}^*, \mathbf{v}^*) = f(\mathbf{x}) + \sum_{i \in I} u_i^* g_i(\mathbf{x}) + \sum_{j=1}^l v_j^* h_j(\mathbf{x}).$$

Its Hessian with respect to \mathbf{x} is:

$$\nabla_{\mathbf{x}}^2 L(\mathbf{x}, \mathbf{u}^*, \mathbf{v}^*) = \nabla^2 f(\mathbf{x}) + \sum_{i \in I} u_i^* \nabla^2 g_i(\mathbf{x}) + \sum_{j=1}^l v_j^* \nabla^2 h_j(\mathbf{x}).$$

To state the second-order conditions, we need to distinguish between strongly and weakly active inequality constraints.

Definition 16.4 (Strongly and Weakly Active Constraints). At a KKT point \mathbf{x}^* with multipliers \mathbf{u}^* :

- $I^+ = \{i \in I : u_i^* > 0\}$ is the set of **strongly active** inequality constraint indices.
- $I^0 = \{i \in I : u_i^* = 0\}$ is the set of **weakly active** inequality constraint indices.

Note that $I = I^+ \cup I^0$ and $I^+ \cap I^0 = \emptyset$.

Definition 16.5 (Critical Cone). The **critical cone** at a KKT point \mathbf{x}^* is defined as:

$$C(\mathbf{x}^*) := \left\{ \mathbf{d} \in \mathbb{R}^n : \begin{array}{ll} \nabla h_j(\mathbf{x}^*)^T \mathbf{d} = 0, & \forall j \in \{1, \dots, l\}, \\ \nabla g_i(\mathbf{x}^*)^T \mathbf{d} = 0, & \forall i \in I^+, \\ \nabla g_i(\mathbf{x}^*)^T \mathbf{d} \leq 0, & \forall i \in I^0. \end{array} \right\}$$

The critical cone generalizes the subspace of first-order feasible variations. Directions in $C(\mathbf{x}^*)$ maintain feasibility with respect to the equality

constraints and strongly active inequality constraints (to first order), while potentially improving weakly active inequality constraints.

16.7.2 Second-Order Necessary Conditions

Theorem 16.5 (KKT Second-Order Necessary Conditions). *Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, and $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j \in \{1, \dots, l\}$, consider the constrained problem*

$$\min_{\mathbf{x} \in X} \left\{ f(\mathbf{x}) : g_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}, h_j(\mathbf{x}) = 0, j \in \{1, \dots, l\} \right\}.$$

Suppose $\nabla g_i(\mathbf{x}^)$, $i \in I$, and $\nabla h_j(\mathbf{x}^*)$, $j \in \{1, \dots, l\}$, are (jointly) linearly independent. If \mathbf{x}^* is a local minimum, then there exist Lagrange multipliers $\mathbf{u}^* \geq 0$ and \mathbf{v}^* such that $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*)$ satisfies the KKT conditions. Additionally,*

$$\mathbf{y}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*) \mathbf{y} \geq 0, \quad \forall \mathbf{y} \in C(\mathbf{x}^*).$$

Remark 16.5 (Special Cases). The second-order necessary conditions reduce to familiar results in special cases:

- **Unconstrained case ($m = l = 0$)**: The critical cone is $C(\mathbf{x}^*) = \mathbb{R}^n$, and the condition becomes $\nabla^2 f(\mathbf{x}^*) \succeq 0$ (positive semidefinite Hessian).
- **Equality constraints only ($m = 0$)**: The critical cone is the null space of the Jacobian of \mathbf{h} , and the condition requires positive semidefiniteness of the restricted Hessian on this subspace.

16.7.3 Second-Order Sufficient Conditions

Theorem 16.6 (KKT Second-Order Sufficient Conditions). *Given a nonempty open set $X \subseteq \mathbb{R}^n$, and functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, m\}$, and $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j \in \{1, \dots, l\}$, consider the constrained problem*

$$\min_{\mathbf{x} \in X} \left\{ f(\mathbf{x}) : g_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}, h_j(\mathbf{x}) = 0, j \in \{1, \dots, l\} \right\}.$$

Suppose \mathbf{x}^* is a KKT point with Lagrange multipliers $\mathbf{u}^* \geq 0$ and \mathbf{v}^* corresponding to the inequality and equality constraints, respectively. If

$$\mathbf{y}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*) \mathbf{y} > 0, \quad \forall \mathbf{y} \in C(\mathbf{x}^*) \setminus \{\mathbf{0}\},$$

then \mathbf{x}^* is a strict local minimum of the above problem.

Remark 16.6 (Gap Between Necessary and Sufficient Conditions). As with unconstrained optimization, there is a gap between the necessary and sufficient second-order conditions. The necessary conditions require the Hessian of the Lagrangian to be positive semidefinite on the critical cone, while the sufficient conditions require strict positive definiteness on the nonzero elements of the critical cone.

16.8 Examples: Applying Second-Order Conditions

Example 16.3 (Finding All KKT Points and Classifying Them). Consider the problem:

$$\min_{\mathbf{x} \in \mathbb{R}^2} \left\{ -0.1(x_1 - 4)^2 + x_2^2 : 1 - x_1^2 - x_2^2 \leq 0 \right\}.$$

Observation: This problem has no global minimum because the objective tends to $-\infty$ as $x_1 \rightarrow +\infty$ with $x_2 = 0$ (which remains feasible for large x_1).

Step 1: Write the KKT conditions.

$$1 - x_1^2 - x_2^2 \leq 0, \tag{16.5}$$

$$\begin{pmatrix} -0.2(x_1 - 4) \\ 2x_2 \end{pmatrix} + u \begin{pmatrix} -2x_1 \\ -2x_2 \end{pmatrix} = \mathbf{0}, \tag{16.6}$$

$$u \geq 0, \tag{16.7}$$

$$u(1 - x_1^2 - x_2^2) = 0. \tag{16.8}$$

Step 2: Consider all cases for the active set I .

Case 1: $I = \emptyset$, i.e., $1 - x_1^2 - x_2^2 < 0$.

Complementary slackness (16.8) implies $u^* = 0$. From (16.6):

$$-0.2(x_1 - 4) = 0 \quad \text{and} \quad 2x_2 = 0,$$

giving $\mathbf{x}^* = (4, 0)^T$. We verify: $1 - 16 - 0 = -15 < 0$, so the constraint is indeed inactive. Thus $\mathbf{x}^* = (4, 0)^T$ with $u^* = 0$ is a KKT point.

Case 2: $I = \{1\}$, i.e., $1 - x_1^2 - x_2^2 = 0$.

The complementary slackness condition (16.8) is automatically satisfied. From (16.6):

$$-0.2(x_1 - 4) - 2ux_1 = 0, \quad 2x_2 - 2ux_2 = 0.$$

The second equation gives $2x_2(1 - u) = 0$, so either $x_2 = 0$ or $u = 1$.

Subcase 2a: $x_2 = 0$. The constraint $1 - x_1^2 = 0$ gives $x_1 = \pm 1$. The first equation becomes:

$$-0.2(x_1 - 4) = 2ux_1 \implies u = \frac{-0.1(x_1 - 4)}{x_1}.$$

For $x_1 = 1$: $u = \frac{-0.1(-3)}{1} = 0.3 > 0$. Valid KKT point: $\mathbf{x}^* = (1, 0)^T$, $u^* = 0.3$.

For $x_1 = -1$: $u = \frac{-0.1(-5)}{-1} = -0.5 < 0$. Not a valid KKT point (violates $u \geq 0$).

Subcase 2b: $u = 1$. The first equation gives:

$$-0.2(x_1 - 4) - 2x_1 = 0 \implies -0.2x_1 + 0.8 - 2x_1 = 0 \implies x_1 = \frac{0.8}{2.2} \approx 0.3636.$$

From the constraint: $x_2^2 = 1 - x_1^2 \approx 1 - 0.132 = 0.868$, so $x_2 \approx \pm 0.9315$.

Valid KKT points: $\mathbf{x}^* \approx (0.3636, \pm 0.9315)^T$, $u^* = 1$.

Step 3: Apply second-order conditions.

Note that at any feasible point, $\nabla g_1(\mathbf{x}^*) = (-2x_1^*, -2x_2^*)^T \neq \mathbf{0}$ (since $(0, 0)$ is not feasible). Therefore, the constraint qualification holds, and the KKT conditions are necessary.

Analysis of $\mathbf{x}^ = (4, 0)^T$, $u^* = 0$:*

Here $I = \emptyset$, so $I^+ = I^0 = \emptyset$ and $C(\mathbf{x}^*) = \mathbb{R}^2$.

The restricted Lagrangian is $L(\mathbf{x}, u^*) = -0.1(x_1 - 4)^2 + x_2^2$, with Hessian:

$$\nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, u^*) = \begin{pmatrix} -0.2 & 0 \\ 0 & 2 \end{pmatrix}.$$

For $\mathbf{y} = (1, 0)^T \in C(\mathbf{x}^*)$:

$$\mathbf{y}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, u^*) \mathbf{y} = -0.2 < 0.$$

The second-order necessary condition fails. Therefore, $\mathbf{x}^* = (4, 0)^T$ is **not a local minimum**.

Analysis of $\mathbf{x}^ = (1, 0)^T$, $u^* = 0.3$:*

Here $I = \{1\}$ and $u^* > 0$, so $I^+ = \{1\}$ and $I^0 = \emptyset$. The critical cone is:

$$C(\mathbf{x}^*) = \{\mathbf{d} \in \mathbb{R}^2 : \nabla g_1(\mathbf{x}^*)^T \mathbf{d} = 0\} = \{\mathbf{d} : (-2, 0)^T \mathbf{d} = 0\} = \{\mathbf{d} : d_1 = 0\}.$$

The restricted Lagrangian Hessian is:

$$\nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, u^*) = \begin{pmatrix} -0.2 & 0 \\ 0 & 2 \end{pmatrix} + 0.3 \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix} = \begin{pmatrix} -0.8 & 0 \\ 0 & 1.4 \end{pmatrix}.$$

For any $\mathbf{y} = (0, y_2)^T \in C(\mathbf{x}^*) \setminus \{\mathbf{0}\}$:

$$\mathbf{y}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, u^*) \mathbf{y} = 1.4 y_2^2 > 0.$$

The second-order sufficient condition is satisfied. Therefore, $\mathbf{x}^* = (1, 0)^T$ is a **strict local minimum**.

Example 16.4 (Second-Order Conditions with Parameter). Consider the problem:

$$\min_{\mathbf{x} \in \mathbb{R}^2} \{(x_1 - 1)^2 + x_2^2 : 2kx_1 - x_2^2 \leq 0\},$$

where $k > 0$ is a positive constant.

The feasible region is the set $\{(x_1, x_2) : x_2^2 \geq 2kx_1\}$, which is the exterior (and boundary) of a parabola opening to the right.

The unconstrained minimum of the objective is at $(1, 0)^T$. Whether this point is feasible depends on the value of k :

- If $k \leq 0$, then $(1, 0)$ is feasible (the constraint $-x_2^2 \leq 0$ is always satisfied).
- If $k > 0$, then $(1, 0)$ is feasible only if $2k \cdot 1 - 0 \leq 0$, i.e., $k \leq 0$. Since we assume $k > 0$, the unconstrained minimum is infeasible.

For $k > 0$, the optimal solution lies on the boundary of the constraint $2kx_1 - x_2^2 = 0$. The KKT conditions and second-order analysis can be used to find and verify the optimal solution, which depends on the value of k .

16.9 Course Overview and Synthesis

This course has covered the fundamental theory and algorithms for nonlinear programming. The following provides a synthesis of the key concepts.

16.9.1 Types of Optima

- **Local minima:** Points where no nearby feasible point has a lower objective value.
- **Strict local minima:** Local minima with strictly greater objective values at all nearby feasible points.
- **Global minima:** Points achieving the lowest objective value over the entire feasible region.

16.9.2 Linear Algebra Foundations

- Linear and affine independence of vectors.
- Positive semidefinite and positive definite matrices.
- Eigenvalues and eigenvectors; their role in characterizing matrix definiteness.

16.9.3 Real Analysis Foundations

- Topological concepts: closed, open, and compact sets; interior and boundary.
- Weierstrass' Theorem: Continuous functions attain their extrema on compact sets.

16.9.4 Convexity Theory

- **Convex sets:** Sets containing line segments between any two of their points.
- **Convex functions:** Functions lying below their chords; characterized by convex epigraphs.
- **Strictly and strongly convex functions:** Providing unique minima and quantitative growth bounds.

- **Operations preserving convexity:** Nonnegative linear combinations, pointwise suprema, composition rules.

16.9.5 Generalizations of Convexity

- **Quasiconvex functions:** Functions with convex sublevel sets.
- **Strictly quasiconvex functions:** Useful for line search algorithms.
- **Pseudoconvex functions:** Functions where stationary points are global minima.

16.9.6 Convex Analysis

- **Carathéodory's theorem:** Points in convex hulls can be expressed using limited vertices.
- **Closest-point theorem:** Unique projections onto closed convex sets.
- **Separation theorems:** Separating and supporting hyperplanes for convex sets.
- **Subgradients:** Generalizing derivatives for nonsmooth convex functions.
- **Characterizations:** First-order (gradient inequality) and second-order (Hessian positive semidefinite) characterizations of differentiable convex functions.

16.9.7 Properties of Convex Optimization Problems

- Local minima are global minima.
- The set of optimal solutions is convex.
- For strictly convex objectives, the optimal solution is unique.

16.9.8 Optimality Conditions

- **Convex problems:** First-order conditions (gradient in normal cone) are necessary and sufficient for global optimality.
- **Unconstrained nonconvex problems:**
 - First-order necessary: $\nabla f(\mathbf{x}^*) = \mathbf{0}$.

- Second-order necessary: $\nabla^2 f(\mathbf{x}^*) \succeq 0$.
- Second-order sufficient: $\nabla f(\mathbf{x}^*) = \mathbf{0}$ and $\nabla^2 f(\mathbf{x}^*) \succ 0$.
- **Equality-constrained problems:** Lagrangian stationarity with restricted Hessian conditions.
- **Inequality-constrained problems:** KKT conditions with complementary slackness.
- **Mixed constraints:** General KKT conditions combining all elements.

16.9.9 Algorithms for Unconstrained Optimization

- **Derivative-free line search:** Golden section, Fibonacci search.
- **Steepest descent:** Using the negative gradient as search direction; linear convergence.
- **Newton's method:** Using second-order information; quadratic local convergence.
- **Conjugate gradient:** Finite convergence for quadratic functions; efficient for large-scale problems.

16.9.10 Algorithms for Constrained Optimization

- **Penalty methods:** Converting constrained to unconstrained problems.
- **Barrier methods:** Interior point approaches for inequality constraints.
- **Sequential quadratic programming:** Solving sequences of QP approximations.
- **Augmented Lagrangian methods:** Combining penalty and Lagrangian approaches.

Introduction to Proof Writing

This appendix provides an introduction to mathematical proof writing, which is essential for understanding and working with the material in this course. Much of optimization theory relies on rigorous mathematical arguments, and developing your proof-writing skills will help you succeed in this course and beyond.

Recommended Reading

- Velleman, D.J. *How to Prove It: A Structured Approach*
- Solow, D. *How to Read and Do Proofs*
- Appendix A: Introduction to Proofs in Real Analysis (this text)
- Appendix B: Proving or Disproving Convexity (this text)

.1 Why Learn Proofs?

Proofs are the foundation of mathematical rigor. In this course, we will use proofs to ensure:

- Properties of sets and functions (e.g., convexity, continuity)
- Correctness of algorithms
- Validity of optimality conditions

Beyond their utility in this course, proof-writing teaches logical thinking and problem-solving skills that are valuable in many areas of engineering, science, and everyday life.

.2 What is a Proof?

A proof is a logical argument that starts with assumptions (given facts) and ends with a conclusion. The key components of a proof are:

1. **Definitions:** Precise meaning of terms used.
2. **Logical steps:** A sequence of reasoning steps, each following logically from previous steps or known facts.
3. **Conclusion:** What you aim to show.

Think of a proof as answering the question: *Why is this true?*

.3 Common Proof Techniques

When asked to prove a statement of the form “ $P \Rightarrow Q$ ” (if P , then Q), there are several standard approaches:

.3.1 Direct Proof

In a direct proof, you assume P is true and use definitions and logical steps to directly show that Q must also be true.

Example .5 (Direct Proof). **Claim:** The intersection of two convex sets is convex.

Proof: Let S_1 and S_2 be convex sets, and let $S = S_1 \cap S_2$. We need to show that S is convex.

Take any $\mathbf{x}_1, \mathbf{x}_2 \in S$ and any $\lambda \in [0, 1]$. Since $\mathbf{x}_1, \mathbf{x}_2 \in S = S_1 \cap S_2$, we have $\mathbf{x}_1, \mathbf{x}_2 \in S_1$ and $\mathbf{x}_1, \mathbf{x}_2 \in S_2$.

By convexity of S_1 : $\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in S_1$.

By convexity of S_2 : $\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in S_2$.

Therefore, $\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in S_1 \cap S_2 = S$, which proves that S is convex. \square

.3.2 Proof by Contraposition

To prove $P \Rightarrow Q$, you can instead prove the logically equivalent statement $\neg Q \Rightarrow \neg P$ (the contrapositive). That is, assume the opposite of what you want to prove and show that it implies the opposite of what you are given.

Example .6 (Proof by Contraposition). **Claim:** If n^2 is even, then n is even.

Proof: We prove the contrapositive: if n is odd, then n^2 is odd. Suppose n is odd. Then $n = 2k + 1$ for some integer k . Therefore,

$$n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1,$$

which is odd. \square

3.3 Proof by Contradiction

Assume both P (what you are given) and $\neg Q$ (the opposite of what you want to prove), then derive a contradiction. This shows that $\neg Q$ cannot hold when P is true.

Example .7 (Proof by Contradiction). **Claim:** $\sqrt{2}$ is irrational.

Proof: Suppose, for contradiction, that $\sqrt{2}$ is rational. Then $\sqrt{2} = p/q$ for some integers p, q with $q \neq 0$ and $\gcd(p, q) = 1$ (i.e., the fraction is in lowest terms).

Squaring both sides: $2 = p^2/q^2$, so $p^2 = 2q^2$.

This means p^2 is even, so p is even (by the previous example). Write $p = 2k$ for some integer k .

Then $(2k)^2 = 2q^2$, so $4k^2 = 2q^2$, which gives $q^2 = 2k^2$.

This means q^2 is even, so q is even.

But then both p and q are even, contradicting our assumption that $\gcd(p, q) = 1$. \square

3.4 Counterexample

If you suspect that a statement $P \Rightarrow Q$ is *false*, you can disprove it by finding a single example where P holds but Q does not.

Example .8 (Counterexample). **Claim:** The set $S = \{\mathbf{x} \in \mathbb{R}_+^2 : x_1x_2 \leq 1\}$ is convex.

Disproof: Consider $\mathbf{y} = (2, 0) \in S$ and $\mathbf{z} = (0, 2) \in S$ (both satisfy the constraint since $2 \cdot 0 = 0 \leq 1$).

Let $\lambda = 0.5$. Then

$$\lambda\mathbf{y} + (1 - \lambda)\mathbf{z} = (1, 1).$$

But $(1)(1) = 1 \leq 1$, so this point is in S .

Let's try different points: $\mathbf{y} = (4, 0)$ and $\mathbf{z} = (0, 4)$. Both satisfy $x_1x_2 = 0 \leq 1$.

With $\lambda = 0.5$: $\lambda\mathbf{y} + (1 - \lambda)\mathbf{z} = (2, 2)$.

But $(2)(2) = 4 > 1$, so $(2, 2) \notin S$.

This counterexample shows S is not convex. \square

.4 Other Proof Techniques

Several other proof techniques are commonly used:

- **Proof by induction:** Used to prove statements about all natural numbers $n \geq n_0$. Prove a base case, then show that if the statement holds for $n = k$, it also holds for $n = k + 1$.
- **Proof by enumeration:** When there are finitely many cases, prove the statement for each case individually.
- **If and only if proofs:** To prove " P if and only if Q " (written $P \Leftrightarrow Q$), prove both directions: $P \Rightarrow Q$ and $Q \Rightarrow P$.
- **Uniqueness proofs:** To show that something is unique, assume there are two such objects and prove they must be equal.
- **Existence proofs:** Constructive proofs exhibit an explicit example; non-constructive proofs show that non-existence leads to a contradiction.

.5 Quantifiers

Quantifiers specify how statements apply to elements of a set:

- **Universal quantifier (\forall):** Means "for all" or "every."
 - Example: $\forall x \in \mathbb{R}, x^2 \geq 0$.
- **Existential quantifier (\exists):** Means "there exists" or "at least one."
 - Example: $\exists x \in \mathbb{R}$ such that $x^2 = 4$.

.5.1 Using Quantifiers in Proofs

Quantifiers determine how to approach a proof:

- **Prove a universal statement** ($\forall x \in S, P(x)$): Start with an *arbitrary* $x \in S$ and show $P(x)$.
- **Disprove a universal statement:** Provide a single counterexample.
- **Prove an existential statement** ($\exists x \in S$ such that $P(x)$):
 - *Constructive:* Find an explicit $x \in S$ satisfying $P(x)$.
 - *Non-constructive:* Use an indirect argument (e.g., Intermediate Value Theorem).

.6 Tips for Writing Clear Proofs

1. **State your proof technique upfront:**
 - “We proceed by contradiction. Suppose that...”
 - “We prove the contrapositive. Assume that...”
2. **Divide the proof into logical parts:** Clearly label assumptions, intermediate steps, and the conclusion. Help the reader follow your reasoning at a glance.
3. **Keep the big picture visible:** The overall strategy should be clear even without reading every detail.
4. **Use lemmas for clarity:** Break long proofs into smaller, manageable pieces. Prove supporting statements as lemmas to maintain readability.
5. **Be precise with quantifiers:** Make clear what is being assumed vs. what is being proven. Specify what “arbitrary” means.
6. **Use mathematical notation consistently:** Define all symbols before using them.

Remark .7. A clear structure makes proofs easier to understand, verify, and remember. With practice, you will develop intuition for choosing the right technique and presenting your arguments effectively.

.7 In-Class Exercise

Example .9 (Exercise). Suppose $S \subseteq \mathbb{R}^n$ is convex. Prove that the set

$$\{Ax + b : x \in S\}$$

is convex for *any* matrix $A \in \mathbb{R}^{m \times n}$ and vector $b \in \mathbb{R}^m$.

Hint: Use the definition of convexity. Let $y_1 = Ax_1 + b$ and $y_2 = Ax_2 + b$ be two points in the image set, and show that their convex combination is also in the image set.

Introduction to Proofs in Real Analysis

This appendix provides examples and exercises pertaining to real analysis concepts, particularly those related to open sets, closed sets, boundaries, and compactness. Most of the arguments below simply formalize intuition—make sure to grasp the intuition first before delving into the details.

Recommended Reading

- Chapter 2 of Bazaraa, Sherali, and Shetty (2006)
- Chapter 4 of Rudin, W. *Principles of Mathematical Analysis*
- Chapter 3 of this text (Real Analysis Foundations)

Recall the definition of an ϵ -neighborhood of a point $\mathbf{x} \in \mathbb{R}^n$:

$$\mathcal{N}_\epsilon(\mathbf{x}) := \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y} - \mathbf{x}\|_2 < \epsilon\}.$$

.8 Classifying Sets: Open, Closed, and Compact

Example .10 (The Entire Space and Empty Set). Consider \mathbb{R}^n and the empty set \emptyset .

Claim: \mathbb{R}^n is both closed and open.

Proof that \mathbb{R}^n is closed: We need to show that $\text{cl}(\mathbb{R}^n) = \mathbb{R}^n$. Any $\mathbf{x} \in \mathbb{R}^n$ is in $\text{cl}(\mathbb{R}^n)$ because every neighborhood $\mathcal{N}_\epsilon(\mathbf{x})$ contains \mathbf{x} itself, so $\mathcal{N}_\epsilon(\mathbf{x}) \cap \mathbb{R}^n \neq \emptyset$. Therefore, $\text{cl}(\mathbb{R}^n) = \mathbb{R}^n$.

(This argument generalizes to show that $S \subseteq \text{cl}(S)$ for any set $S \subseteq \mathbb{R}^n$.)

Proof that \mathbb{R}^n is open: We need to show that $\text{int}(\mathbb{R}^n) = \mathbb{R}^n$. Any $\mathbf{x} \in \mathbb{R}^n$ is in $\text{int}(\mathbb{R}^n)$ because the 1-neighborhood $\mathcal{N}_1(\mathbf{x})$ is entirely

contained within \mathbb{R}^n . Therefore, $\text{int}(\mathbb{R}^n) = \mathbb{R}^n$.

(Note that $\text{int}(S) \subseteq S$ for any set $S \subseteq \mathbb{R}^n$.)

Since $\text{cl}(\mathbb{R}^n) = \text{int}(\mathbb{R}^n) = \mathbb{R}^n$, the boundary is:

$$\partial\mathbb{R}^n = \text{cl}(\mathbb{R}^n) \setminus \text{int}(\mathbb{R}^n) = \mathbb{R}^n \setminus \mathbb{R}^n = \emptyset.$$

Note that \mathbb{R}^n is *not* compact because it is not bounded.

Remark .8. The only subsets of \mathbb{R}^n that are *both* closed and open are \mathbb{R}^n and \emptyset .

.9 The Half-Open Box

Example .11 (The Set $[\mathbf{0}, \mathbf{1})$). Consider the set $S = [\mathbf{0}, \mathbf{1}) \subset \mathbb{R}^n$, where $\mathbf{1}$ denotes a vector with all components equal to one.

Claim: S is neither closed nor open.

Proof that S is not closed: We show that $\mathbf{1} \in \text{cl}(S)$. Since $\mathbf{1} \notin S$, this implies $\text{cl}(S) \neq S$.

For any $\epsilon > 0$, the ϵ -neighborhood of $\mathbf{1}$ contains the vector

$$\mathbf{y} := \left(\max \left\{ 0, 1 - \frac{\epsilon}{2\sqrt{n}} \right\}, \dots, \max \left\{ 0, 1 - \frac{\epsilon}{2\sqrt{n}} \right\} \right) \in S.$$

Therefore, $\mathcal{N}_\epsilon(\mathbf{1}) \cap S \neq \emptyset$ for each $\epsilon > 0$, so $\mathbf{1} \in \text{cl}(S)$.

Since S is not closed, it cannot be compact.

Proof that S is not open: We show that $\mathbf{0} \notin \text{int}(S)$. Since $\mathbf{0} \in S$, this implies $S \neq \text{int}(S)$.

Any ϵ -neighborhood of $\mathbf{0}$ contains points not in S . Specifically, $\mathcal{N}_\epsilon(\mathbf{0})$ contains:

$$\mathbf{y} := \left(-\frac{\epsilon}{2\sqrt{n}}, \dots, -\frac{\epsilon}{2\sqrt{n}} \right) \notin S.$$

Therefore, $\mathbf{0} \notin \text{int}(S)$.

Closure: We have $\text{cl}(S) = [\mathbf{0}, \mathbf{1}]$.

Interior: $\text{int}(S) = (\mathbf{0}, \mathbf{1})$.

Boundary: $\partial S = [\mathbf{0}, \mathbf{1}] \setminus (\mathbf{0}, \mathbf{1}) = \{\mathbf{x} \in [\mathbf{0}, \mathbf{1}] : x_i \in \{0, 1\} \text{ for at least one } i\}$.

.10 The Unit Ball and Unit Sphere

Example .12 (The Unit Ball). Consider the unit ball $S = \{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 \leq 1\}$.

Exercise: Show that S is closed and compact with:

$$\text{int}(S) = \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 < 1 \right\},$$

$$\text{cl}(S) = S,$$

$$\partial S = \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 = 1 \right\}.$$

Hints:

- To show $\text{int}(S) = \{\mathbf{x} : \|\mathbf{x}\|^2 < 1\}$, first show that for any point in the open ball, a sufficiently small neighborhood is entirely contained in S . Then show that for any point on the boundary ($\|\mathbf{x}\|^2 = 1$), every neighborhood contains points outside S .
- To show $\text{cl}(S) = S$, show that for any $\mathbf{x} \notin S$, a sufficiently small neighborhood does not intersect S .

Example .13 (The Unit Sphere). Consider the unit sphere $S = \{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 = 1\}$.

Exercise: Show that S is closed and compact with:

$$\text{int}(S) = \emptyset,$$

$$\text{cl}(S) = S,$$

$$\partial S = S.$$

Hint: Any neighborhood of a point on the sphere contains points both inside and outside the unit ball.

.11 The Unit Box

Example .14 (The Unit Box). Consider $S = \{\mathbf{x} \in \mathbb{R}^n : \max_{i \in [n]} |x_i| \leq 1\}$.

We can rewrite S as:

$$S = \bigcap_{i=1}^n \{\mathbf{x} \in \mathbb{R}^n : -1 \leq x_i \leq 1\}.$$

Since each set $\{\mathbf{x} \in \mathbb{R}^n : -1 \leq x_i \leq 1\}$ is closed, and the intersection of closed sets is closed, S is closed.

Since S is bounded (contained in the ball of radius \sqrt{n}) and closed, S is compact.

We have:

$$\text{cl}(S) = S,$$

$$\text{int}(S) = \{\mathbf{x} \in \mathbb{R}^n : -1 < x_i < 1, i = 1, \dots, n\} = (-1, 1),$$

$$\partial S = \{\mathbf{x} \in [-1, 1]^n : x_i \in \{-1, 1\} \text{ for at least one } i\}.$$

.12 A Lower-Dimensional Set

Example .15 (A Square in \mathbb{R}^3). Consider $S = \{\mathbf{x} \in \mathbb{R}^3 : -1 \leq x_1, x_2 \leq 1, x_3 = 0\}$.

Exercise: Show that S is closed and compact with:

$$\text{int}(S) = \emptyset,$$

$$\text{cl}(S) = S,$$

$$\partial S = S.$$

Hint: Any neighborhood of $\mathbf{x} \in S$ contains points with $y_3 \neq 0$, which are not in S .

.13 Useful Theorems for Proving Openness and Closedness

The following powerful results can greatly simplify proofs about open and closed sets.

Theorem .7 (Pre-image of Open Sets). *Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous function. Then for any open set $S \subseteq \mathbb{R}$, the pre-image*

$$f^{-1}(S) := \{\mathbf{y} \in \mathbb{R}^n : f(\mathbf{y}) \in S\}$$

is open.

Corollary .8 (Pre-image of Closed Sets). *Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous function. Then for any closed set $S \subseteq \mathbb{R}$, the pre-image $f^{-1}(S)$ is closed.*

Exercises: Use the above results to prove the following:

1. The set $\{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 < 1\}$ is open.

Hint: Consider $f(\mathbf{x}) = \sum_{i=1}^n x_i^2$ and $S = (-\infty, 1)$.

2. The following sets are closed:

- $\{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 \leq 1\}$ and $\{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 = 1\}$
- $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^\top \mathbf{x} \leq b\}$ and $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^\top \mathbf{x} = b\}$ for some $\mathbf{a} \in \mathbb{R}^n$, $b \in \mathbb{R}$
- $\{\mathbf{x} \in \mathbb{R}^n : \max_{i \in [n]} |x_i| \leq 1\}$
- The singleton set $\{\bar{\mathbf{x}}\}$ for any $\bar{\mathbf{x}} \in \mathbb{R}^n$

Proving or Disproving Convexity of Sets and Functions

This appendix provides sample proofs and counterexamples for establishing (or disproving) convexity of sets and functions. These examples illustrate the proof techniques introduced in Appendix A applied to core concepts from convex analysis.

Recommended Reading

- Chapter 2 and 3 of Bazaraa, Sherali, and Shetty (2006)
- Chapter 3 of Boyd and Vandenberghe (2004)
- Chapters 4–5 of this text (Convex Functions and Subgradients)

.14 Proving Convexity of Sets

Recall that a set $S \subseteq \mathbb{R}^n$ is **convex** if for any $\mathbf{y}, \mathbf{z} \in S$ and any $\lambda \in [0, 1]$:

$$\lambda\mathbf{y} + (1 - \lambda)\mathbf{z} \in S.$$

Example .16 (The Unit Ball is Convex). **Claim:** The set $S = \{\mathbf{x} \in \mathbb{R}^2 : x_1^2 + x_2^2 \leq 1\}$ is convex.

Proof: Consider any $\mathbf{y}, \mathbf{z} \in S$ and any $\lambda \in [0, 1]$. Since $\mathbf{y}, \mathbf{z} \in S$:

$$y_1^2 + y_2^2 \leq 1, \tag{9}$$

$$z_1^2 + z_2^2 \leq 1. \tag{10}$$

Note that $\lambda\mathbf{y} + (1 - \lambda)\mathbf{z} = (\lambda y_1 + (1 - \lambda)z_1, \lambda y_2 + (1 - \lambda)z_2)$.

We need to show:

$$(\lambda y_1 + (1 - \lambda)z_1)^2 + (\lambda y_2 + (1 - \lambda)z_2)^2 \leq 1.$$

We compute:

$$\begin{aligned} & (\lambda y_1 + (1 - \lambda)z_1)^2 + (\lambda y_2 + (1 - \lambda)z_2)^2 \\ &= \lambda^2(y_1^2 + z_1^2) + (1 - \lambda)^2(z_1^2 + z_2^2) + 2\lambda(1 - \lambda)(y_1z_1 + y_2z_2) \\ &\leq \lambda^2 \cdot 1 + (1 - \lambda)^2 \cdot 1 + 2\lambda(1 - \lambda)(y_1z_1 + y_2z_2) \\ &\leq \lambda^2 + (1 - \lambda)^2 + \lambda(1 - \lambda)(y_1^2 + z_1^2 + y_2^2 + z_2^2) \\ &\leq \lambda^2 + (1 - \lambda)^2 + 2\lambda(1 - \lambda) \\ &= (\lambda + (1 - \lambda))^2 = 1, \end{aligned}$$

where we used (9)–(10) and the inequality $2ab \leq a^2 + b^2$. \square

Remark .9. This argument generalizes to show that $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 \leq 1\}$ is convex.

Example .17 (Polyhedra are Convex). **Claim:** The set $S = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is convex.

Proof: Consider any $\mathbf{y}, \mathbf{z} \in S$ and any $\lambda \in [0, 1]$. Since $\mathbf{y}, \mathbf{z} \in S$:

$$\begin{aligned} A\mathbf{y} &= \mathbf{b}, \quad \mathbf{y} \geq \mathbf{0}, \\ A\mathbf{z} &= \mathbf{b}, \quad \mathbf{z} \geq \mathbf{0}. \end{aligned}$$

We need to show $A(\lambda\mathbf{y} + (1 - \lambda)\mathbf{z}) = \mathbf{b}$ and $\lambda\mathbf{y} + (1 - \lambda)\mathbf{z} \geq \mathbf{0}$.

For the equality constraint:

$$\begin{aligned} A(\lambda\mathbf{y} + (1 - \lambda)\mathbf{z}) &= \lambda(A\mathbf{y}) + (1 - \lambda)(A\mathbf{z}) \\ &= \lambda\mathbf{b} + (1 - \lambda)\mathbf{b} = \mathbf{b}. \end{aligned}$$

For non-negativity: Since $\mathbf{y} \geq \mathbf{0}$, $\mathbf{z} \geq \mathbf{0}$, and $\lambda \in [0, 1]$:

$$\lambda\mathbf{y} + (1 - \lambda)\mathbf{z} \geq \mathbf{0}.$$

Therefore, $\lambda\mathbf{y} + (1 - \lambda)\mathbf{z} \in S$. \square

Remark .10. This proof actually shows that $\{\mathbf{x} : A\mathbf{x} = \mathbf{b}\}$ and $\{\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}$ are each convex. Since intersections of convex sets are convex (see Chapter 2), this provides an alternative proof.

Example .18 (A Non-Obvious Convex Set). **Claim:** The set $S = \{\mathbf{x} \in \mathbb{R}^2 : x_1^2 - x_1x_2^2 + x_2^4 + 1 \geq 0\}$ is convex.

Proof: We show that $S = \mathbb{R}^2$, which is trivially convex.

For any $\mathbf{y} \in \mathbb{R}^2$:

$$y_1^2 - y_1y_2^2 + y_2^4 + 1 = \left(y_1 - \frac{1}{2}y_2^2\right)^2 + \frac{3}{4}y_2^4 + 1.$$

Since each term on the right is non-negative, the entire expression is at least 1, which is positive. Therefore, every $\mathbf{y} \in \mathbb{R}^2$ satisfies the constraint, so $S = \mathbb{R}^2$. \square

Remark .11. This example illustrates that sets defined by polynomial inequalities are not always easy to analyze. The trick of completing the square revealed that this set is actually the entire space.

.15 Disproving Convexity of Sets

To disprove convexity, we only need to find *one* counterexample: points $\mathbf{y}, \mathbf{z} \in S$ and $\lambda \in [0, 1]$ such that $\lambda\mathbf{y} + (1 - \lambda)\mathbf{z} \notin S$.

Example .19 (A Non-Convex Set). **Claim:** The set $S = \{\mathbf{x} \in \mathbb{R}_+^2 : x_1x_2 = 0\}$ is *not* convex.

Proof: Let $\mathbf{y} = (1, 0)$ and $\mathbf{z} = (0, 1)$. Note that $\mathbf{y}, \mathbf{z} \in S$ since $y_1y_2 = 0$ and $z_1z_2 = 0$.

Let $\lambda = 0.5$. Then:

$$\mathbf{w} := \lambda\mathbf{y} + (1 - \lambda)\mathbf{z} = (0.5, 0.5).$$

But $w_1w_2 = 0.5 \times 0.5 = 0.25 \neq 0$, so $\mathbf{w} \notin S$.

This counterexample proves S is not convex. \square

.16 Proving Convexity of Functions

Recall that a function $f : \mathcal{D} \rightarrow \mathbb{R}$ is **convex** if \mathcal{D} is convex and for any $\mathbf{y}, \mathbf{z} \in \mathcal{D}$ and any $\lambda \in [0, 1]$:

$$f(\lambda\mathbf{y} + (1 - \lambda)\mathbf{z}) \leq \lambda f(\mathbf{y}) + (1 - \lambda)f(\mathbf{z}).$$

Example .20 (Quadratic Functions). **Claim:** The function $f(\mathbf{x}) = x_1^2 + x_2^2$ is convex on \mathbb{R}^2 .

Proof: The domain \mathbb{R}^2 is convex. For any $\mathbf{y}, \mathbf{z} \in \mathbb{R}^2$ and $\lambda \in [0, 1]$:

$$\begin{aligned} f(\lambda\mathbf{y} + (1 - \lambda)\mathbf{z}) &= (\lambda y_1 + (1 - \lambda)z_1)^2 + (\lambda y_2 + (1 - \lambda)z_2)^2 \\ &= \lambda^2(y_1^2 + y_2^2) + (1 - \lambda)^2(z_1^2 + z_2^2) + 2\lambda(1 - \lambda)(y_1z_1 + y_2z_2) \\ &\leq \lambda^2(y_1^2 + y_2^2) + (1 - \lambda)^2(z_1^2 + z_2^2) + \lambda(1 - \lambda)(y_1^2 + z_1^2 + y_2^2 + z_2^2) \\ &= (\lambda^2 + \lambda(1 - \lambda))(y_1^2 + y_2^2) + ((1 - \lambda)^2 + \lambda(1 - \lambda))(z_1^2 + z_2^2) \\ &= \lambda(y_1^2 + y_2^2) + (1 - \lambda)(z_1^2 + z_2^2) \\ &= \lambda f(\mathbf{y}) + (1 - \lambda)f(\mathbf{z}), \end{aligned}$$

where we used $2ab \leq a^2 + b^2$. □

Remark .12. Notice the similarity between this proof and Example 1 (convexity of the unit ball). This connection between convex functions and convex sets is explored in Chapter 4.

Example .21 (Linear Functions are Both Convex and Concave).

Claim: The function $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x} + b$ is both convex and concave on \mathbb{R}^n .

Proof: For any $\mathbf{y}, \mathbf{z} \in \mathbb{R}^n$ and $\lambda \in [0, 1]$:

$$\begin{aligned} f(\lambda\mathbf{y} + (1 - \lambda)\mathbf{z}) &= \mathbf{a}^\top(\lambda\mathbf{y} + (1 - \lambda)\mathbf{z}) + b \\ &= \lambda\mathbf{a}^\top\mathbf{y} + (1 - \lambda)\mathbf{a}^\top\mathbf{z} + b \\ &= \lambda(\mathbf{a}^\top\mathbf{y} + b) + (1 - \lambda)(\mathbf{a}^\top\mathbf{z} + b) \\ &= \lambda f(\mathbf{y}) + (1 - \lambda)f(\mathbf{z}). \end{aligned}$$

Since we have equality (not just \leq or \geq), f is both convex and concave. □

Example .22 (The Logarithm is Concave). **Claim:** The function $f(x) = \ln(x)$ is concave on $(0, +\infty)$.

Proof: The domain $(0, +\infty)$ is convex. For any $y, z \in (0, +\infty)$ and $\lambda \in [0, 1]$, we need to show:

$$\ln(\lambda y + (1 - \lambda)z) \geq \lambda \ln(y) + (1 - \lambda) \ln(z) = \ln(y^\lambda z^{1-\lambda}).$$

Since \ln is increasing, this is equivalent to:

$$\lambda y + (1 - \lambda)z \geq y^\lambda z^{1-\lambda}.$$

This is the weighted AM-GM inequality, which states that the weighted arithmetic mean is at least the weighted geometric mean. \square

.17 Disproving Convexity of Functions

To disprove convexity, find points \mathbf{y}, \mathbf{z} and $\lambda \in [0, 1]$ such that:

$$f(\lambda \mathbf{y} + (1 - \lambda)\mathbf{z}) > \lambda f(\mathbf{y}) + (1 - \lambda)f(\mathbf{z}).$$

Similarly, to disprove concavity, find points where the opposite strict inequality holds.

Example .23 (A Function That is Neither Convex Nor Concave).

Claim: The function $f(\mathbf{x}) = x_1 x_2$ is neither convex nor concave on \mathbb{R}^2 .

Disproving convexity: Let $\mathbf{y} = (1, 0)$, $\mathbf{z} = (0, 1)$, and $\lambda = 0.5$. Then $\lambda \mathbf{y} + (1 - \lambda)\mathbf{z} = (0.5, 0.5)$, and:

$$f(\lambda \mathbf{y} + (1 - \lambda)\mathbf{z}) = 0.5 \times 0.5 = 0.25 > 0 = \lambda f(\mathbf{y}) + (1 - \lambda)f(\mathbf{z}).$$

Since the inequality goes the wrong way, f is not convex.

Disproving concavity: Let $\mathbf{u} = (-1, 0)$, $\mathbf{v} = (0, 1)$, and $\theta = 0.5$. Then $\theta \mathbf{u} + (1 - \theta)\mathbf{v} = (-0.5, 0.5)$, and:

$$f(\theta \mathbf{u} + (1 - \theta)\mathbf{v}) = -0.5 \times 0.5 = -0.25 < 0 = \theta f(\mathbf{u}) + (1 - \theta)f(\mathbf{v}).$$

Since the inequality goes the wrong way, f is not concave. \square

.18 Operations Preserving Convexity

In later chapters, we develop systematic tools for proving convexity without resorting to the definition each time. Key operations that preserve convexity include:

- Non-negative weighted sums of convex functions
- Composition with affine mappings
- Pointwise maximum of convex functions
- Partial minimization over convex sets

These tools, combined with a library of basic convex functions, greatly simplify convexity proofs in practice.

Remark .13 (Exercise). Prove the lemma on operations preserving convexity of sets from Chapter 2. Some of these may appear in homework assignments.