OPTIMAL QUADRATIC
PROGRAMMING ALGORITHMS
Aims and Scope
Optimization has been expanding in all directions at an astonishing rate during the last few decades. New algorithmic and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace, and our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in all areas of applied mathematics, engineering, medicine, economics and other sciences.

The series *Springer Optimization and Its Applications* publishes undergraduate and graduate textbooks, monographs and state-of-the-art expository works that focus on algorithms for solving optimization problems and also study applications involving such problems. Some of the topics covered include nonlinear optimization (convex and nonconvex), network flow problems, stochastic optimization, optimal control, discrete optimization, multi-objective programming, description of software packages, approximation techniques and heuristic approaches.
OPTIMAL QUADRATIC PROGRAMMING ALGORITHMS

With Applications to Variational Inequalities

By

ZDENĚK DOSTÁL
VŠB - Technical University of Ostrava, Czech Republic

Springer
To Maruška, Matěj, and Michal, the dearest ones
Preface

The main purpose of this book is to present some recent results concerning the development of in a sense optimal algorithms for the solution of large bound and/or equality constrained quadratic programming (QP) problems. The unique feature of these algorithms is the rate of convergence in terms of the bounds on the spectrum of the Hessian matrix of the cost function. If applied to the class of QP problems with the cost functions whose Hessian has the spectrum confined to a given positive interval, the algorithms can find approximate solutions in a uniformly bounded number of simple iterations, such as the matrix–vector multiplications. Moreover, if the class of problems admits a sparse representation of the Hessian, it simply follows that the cost of the solution is proportional to the number of unknowns.

Notice also that the cost of duplicating the solution is proportional to the number of variables. The only difference is a constant coefficient. But the constants are important; people are interested in their salaries, as Professor Babuška nicely points out. We therefore tried hard to present a quantitative theory of convergence of our algorithms wherever possible. In particular, we tried to give realistic bounds on the rate of convergence, usually in terms of the extreme nonzero eigenvalues of the matrices involved in the definition of the problem. The theory covers also the problems with dependent constraints.

The presentation of each new algorithm is complete in the sense that it starts from its classical predecessors, describes their drawbacks, introduces modifications that improve their performance, and documents the improvements by numerical experiments. Since the exposition is self-contained, the book can serve as an introductory text for anybody interested in QP. Moreover, since the solution of a number of more general nonlinear problems can be reduced to the solution of a sequence of QP problems, the book can also serve as a convenient introduction to nonlinear programming. Such presentation has also a considerable methodological appeal as it enables us to separate the simple geometrical ideas, which are behind many theoretical results and algorithms, from the technical difficulties arising in the analysis of more general nonlinear optimization problems.
Our algorithms are based on modifications of the active set strategy that is optionally combined with variants of the augmented Lagrangian method. Small observations and careful analysis resulted in their qualitatively improved performance. Surprisingly, these methods can solve some large QP problems with less effort than a single step of the popular interior point methods. The reason is that the standard implementation of the interior point methods can hardly use a favorable distribution of the spectrum of the Hessian due to the barrier function. On the other hand, the standard implementations of interior point methods do not rely on the conditioning of the Hessian and can exploit efficiently its sparsity pattern to simplify LU decomposition. Hence there are also many problems that can be solved more efficiently by the interior point methods, and our approach may be considered as complementary to them.

Contact Problems and Scalable Algorithms

The development of the algorithms presented in this book was motivated by an effort to solve the large sparse problems arising from the discretization of elliptic variational inequalities, such as those describing the equilibrium of elastic bodies in mutual contact. A simple academic example is the contact problem of elasticity to describe the deformation and contact pressure due to volume forces of the cantilever cube over the obstacle in Fig. 0.1.

Fig. 0.1. Cantilever cube over the obstacle

The class of problems arising from various discretizations of a given variational inequality by the finite element or boundary element method can be reduced to the class of QP problems with a uniformly bounded spectrum by an application of the FETI/BETI (Finite/Boundary Element Tearing and Interconnecting)-based domain decomposition methods. Let us recall that the basic idea of these methods is to decompose the domain into subdomains as in Fig. 0.2 and then “glue” them by the Lagrange multipliers that are found by an iterative procedure.
Combination of the results on scalability of variants of the FETI methods for unconstrained problems with the algorithms presented in this book resulted in development of scalable algorithms for elliptic boundary variational inequalities. Let us recall that an algorithm is numerically scalable if the cost of the solution is nearly proportional to the number of unknowns, and it enjoys the parallel scalability if the time required for the solution can be reduced nearly proportionally to the number of available processors. For example, the solution of our toy problem required from 111 to 133 sparse matrix multiplications for varying discretizations with the number of nodes on the surface ranging from 417 to 163275.

As a more realistic example, let us consider the problem to describe the deformation and contact pressure in the ball bearings in Fig. 0.4. We can easily recognize that it comprises several bodies – balls, rings, and cages. The balls are not fixed in their cages, so that their stiffness matrices are necessarily singular and the discretized nonpenetration conditions can be described naturally by dependent constraints. Though the displacements and forces are typically given on parts of the surfaces of some bodies, exact places where the deformed balls come into contact with the cages or the rings are known only after the problem is solved.
It should not be surprising that the duality-based methods can be more successful for the solution of variational inequalities than for the linear problems. The duality turns the general inequality constraints into bound constraints for free; the aspect not exploited in the solution of linear problems. The first fully scalable algorithm for numerical solution of linear problems, FETI, was introduced only in the early 1990s. It was quite challenging to get similar results for variational inequalities. Since the cost of the solution of a linear problem is proportional to the number of variables, a scalable algorithm must identify the active constraints in a sense for free!

Synopsis of the Book

The book is arranged into three parts. We start the introductory part by reviewing some well-known facts on linear algebra in the form that is useful in the following analysis, including less standard estimates, matrix decompositions of semidefinite matrices with known kernel, and spectral theory. The results are then used in the review of standard results on convex and quadratic programming. Though many results concerning the existence and uniqueness of QP problems are special cases of more general theory of nonlinear programming, it is often possible to develop more straightforward proofs that exploit specific structure of the QP problems, in particular the three-term Taylor’s expansion, and sometimes to get stronger results. We paid special attention to the results for dependent constraints and/or positive semidefinite Hessian, including the sensitivity analysis and the duality theory in Sect. 2.6.5.

The second part is the core of the book and comprises four sections on the algorithms for specific types of constraints. It starts with Chap. 3 which summarizes the basic facts on the application of the conjugate gradient method to unconstrained QP problems. The material included is rather standard, possibly except Sect. 3.7 on the preconditioning by a conjugate projector.

Chapter 4 reviews in detail the Uzawa-type algorithms. A special attention is paid to the quantitative analysis of the penalty method and of an inexact solution of auxiliary unconstrained problems. The standard results on exact algorithms are also included. A kind of optimality is proved for a variant of the inexact penalty method and for the semimonotonic augmented Lagrangian algorithm SMALE. A bound on the penalty parameter which guarantees the linear convergence is also presented.

Chapter 5 describes the adaptations of the conjugate gradient algorithm to the solution of bound constrained problems. The algorithms include the variants of Polyak’s algorithm with the inexact solution of auxiliary problems and the precision control which preserves the finite termination property. The main result of this chapter is the MPRGP algorithm with the linear rate of convergence which depends on the extreme eigenvalues of the Hessian of the cost function. We show that the rate of convergence can be improved by the preconditioning exploiting the conjugate projectors.
The last chapter of the second part combines Chaps. 4 and 5 to obtain optimal convergence results for the SMALBE algorithm for the solution of bound and equality constrained QP problems.

The performance of the representative algorithms of the second part is illustrated in each chapter by numerical experiments. We chose the benchmarks arising from the discretization of the energy functions associated with the Laplace operator to mimic typical applications. The benchmarks involve in each chapter one ill-conditioned problem to illustrate the typical performance of our algorithms in such situation and the class of well-conditioned problems to demonstrate the optimality of the best algorithms. Using the same cost functions in all benchmarks of the second part in combination with the boundary inequalities and multipoint constraints enables additional comparison. For convenience of the reader, Chaps. 3–5 are introduced by an overview of the algorithms presented there.

The concept of optimality is fully exploited in the last part of our book, where the algorithms of Chaps. 5 and 6 are combined with the FETI–DP (Dual–Primal FETI) and TFETI (Total FETI) methods to develop theoretically supported scalable algorithms for numerical solution of the classes of problems arising from the discretization of elliptic boundary variational inequalities. The numerical and parallel scalability is demonstrated on the solution of a coercive boundary variational inequality and on the solution of a semicoercive multidomain problem with more then two million nodal variables. The application of the algorithms presented in the last part of our book to the solution of contact problems of elasticity in two and three dimensions, including the contact problems with friction, is straightforward. The same is true for the applications of the algorithms to the development of scalable BETI-based algorithms for the solution of contact problems discretized by the direct boundary element methods. An interested reader can find the references at the end of the last two chapters.

Acknowledgments

Most of the nonstandard results presented in this book have been found by the author over the last fifteen years, often in cooperation with other colleagues. I would like to acknowledge here my thanks especially to Ana Friedlander and Mario Martínez for proper assessment of the efficiency of the augmented Lagrangian methods, to Sandra A. Santos and F.A.M. Gomes for their share in early development of our algorithms for the solution of variational inequalities, to Joachim Schöberl for sharing his original insight into the gradient projection method, to Dan Stefanica for joint work on scalable FETI–DP methods, especially for the proofs of optimal estimates without preconditioners, and to Charbel Farhat for drawing attention to practical aspects of our algorithms and an inspiration for thinking twice about simple topics.

The first results on optimal algorithms were presented at the summer schools organized by Ivo Marek, whose encouragement was essential in the decision to write this book.
My thanks go also to my colleagues and students from the Faculty of Electrical Engineering and Computer Science of VŠB–Technical University of Ostrava. Vít Vondrák implemented the first versions of the algorithms to the solution of contact problems of mechanics and shape optimization, David Horák first implemented many variants of the algorithms that appear in this book, and Dalibor Lukáš adapted the algorithms of Chap. 4 to the solution of the Stokes problem. My thanks go also to Marta Domorádová for her share in research of conjugate projectors and assistance with figures, to Radek Kučera who adapted the algorithms for bound constrained QP to the solution of more general problems with separated constraints and carried out a lot of joint work, and to Petr Beremlijski, Tomáš Kozubek, and Oldřich Vlach who applied at least some of these algorithms to the solution of engineering benchmarks. The book would be much worse without critical reading of its early versions by the colleagues mentioned above and especially by Marie Sadowská, who also participated with Jiří Bouchala in development of scalable algorithms for the problems discretized by boundary elements. Marta Domorádová, Marie Sadowská, Dalibor Lukáš, and David Horák kindly assisted with numerical experiments. There would be more errors in English if it were not for Barunka Dostálová.

It was a pleasure to work on the book with the publication staff at Springer. I am especially grateful to Elizabeth Loew and Frank Ganz for their share in final refinements of the book.

I gratefully acknowledge the support by the grants of the Ministry of Education of the Czech Republic No. MSM6198910027, GA CR 201/07/0294, and AS CR 1ET400300415. Last, but not least, my thanks go to the VŠB–Technical University of Ostrava and to the Faculty of Electrical Engineering and Computer Science for supporting the research of the whole group when needed.

The book is inscribed to my closest family, who have never complained much when my mind switched to quadratic forms. I would have hardly finished the book without the kind support of my wife Maruška.

Ostrava and Dolní Bečva

Zdeněk Dostál

August 2008
## Contents

Preface ........................................................ v ii  

Part I Background  

1 Linear Algebra ......................................................... 3  
1.1 Vectors .................................................................. 3  
1.2 Matrices and Matrix Operations ................................. 5  
1.3 Matrices and Mappings .............................................. 6  
1.4 Inverse and Generalized Inverse Matrices ................. 8  
1.5 Direct Methods for Solving Linear Equations ............. 9  
1.6 Norms .................................................................. 12  
1.7 Scalar Products ...................................................... 14  
1.8 Eigenvalues and Eigenvectors ................................. 17  
1.9 Matrix Decompositions ........................................... 19  
1.10 Penalized Matrices ............................................... 22  

2 Optimization ............................................................ 27  
2.1 Optimization Problems and Solutions ....................... 27  
2.2 Unconstrained Quadratic Programming ..................... 28  
2.2.1 Quadratic Cost Functions .................................. 28  
2.2.2 Unconstrained Minimization of Quadratic Functions . 29  
2.3 Convexity ............................................................ 31  
2.3.1 Convex Quadratic Functions ............................... 32  
2.3.2 Local and Global Minimizers of Convex Function ... 34  
2.3.3 Existence of Minimizers ..................................... 35  
2.3.4 Projections to Convex Sets ................................. 36  
2.4 Equality Constrained Problems ............................... 38  
2.4.1 Optimality Conditions ....................................... 39  
2.4.2 Existence and Uniqueness .................................. 41  
2.4.3 KKT Systems .................................................... 42
2.4.4 Min-max, Dual, and Saddle Point Problems .................. 44
2.4.5 Sensitivity ............................................ 46
2.4.6 Error Analysis ......................................... 47
2.5 Inequality Constrained Problems ................................. 49
2.5.1 Polyhedral Sets ........................................ 49
2.5.2 Farkas’s Lemma ......................................... 50
2.5.3 Necessary Optimality Conditions for Local Solutions .... 51
2.5.4 Existence and Uniqueness ................................ 52
2.5.5 Optimality Conditions for Convex Problems ............... 54
2.5.6 Optimality Conditions for Bound Constrained Problems 55
2.5.7 Min-max, Dual, and Saddle Point Problems ............... 55
2.6 Equality and Inequality Constrained Problems .................. 57
2.6.1 Optimality Conditions ..................................... 58
2.6.2 Existence and Uniqueness ................................ 59
2.6.3 Partially Bound and Equality Constrained Problems ... 59
2.6.4 Duality for Dependent Constraints .......................... 61
2.6.5 Duality for Semicoercive Problems ........................ 64
2.7 Linear Programming ........................................... 69
2.7.1 Solvability and Localization of Solutions ................... 69
2.7.2 Duality in Linear Programming ............................. 70

Part II Algorithms

3 Conjugate Gradients for Unconstrained Minimization ......... 73
3.1 Conjugate Directions and Minimization ....................... 74
3.2 Generating Conjugate Directions and Krylov Spaces ....... 77
3.3 Conjugate Gradient Method .................................. 78
3.4 Restarted CG and the Gradient Method ....................... 81
3.5 Rate of Convergence and Optimality .......................... 82
3.5.1 Min-max Estimate ....................................... 82
3.5.2 Estimate in the Condition Number ....................... 84
3.5.3 Convergence Rate of the Gradient Method ............... 86
3.5.4 Optimality ............................................. 87
3.6 Preconditioned Conjugate Gradients ........................... 87
3.7 Preconditioning by Conjugate Projector ....................... 90
3.7.1 Conjugate Projectors .................................... 90
3.7.2 Minimization in Subspace ................................ 91
3.7.3 Conjugate Gradients in Conjugate Complement .......... 92
3.7.4 Preconditioning Effect .................................. 94
3.8 Conjugate Gradients for More General Problems ............ 96
3.9 Convergence in Presence of Rounding Errors ................. 97
3.10 Numerical Experiments ..................................... 98
3.10.1 Basic CG and Preconditioning ........................... 98
3.10.2 Numerical Demonstration of Optimality .................. 99
4 Equality Constrained Minimization ........................................... 103
4.1 Review of Alternative Methods ........................................... 105
4.2 Penalty Method .................................................................. 107
  4.2.1 Minimization of Augmented Lagrangian ..................... 108
  4.2.2 An Optimal Feasibility Error Estimate for
        Homogeneous Constraints ...................................... 109
  4.2.3 Approximation Error and Convergence ...................... 111
  4.2.4 Improved Feasibility Error Estimate ....................... 112
  4.2.5 Improved Approximation Error Estimate ................. 113
  4.2.6 Preconditioning Preserving Gap in the Spectrum .......... 115
4.3 Exact Augmented Lagrangian Method ................................. 116
  4.3.1 Algorithm ...................................................... 117
  4.3.2 Convergence of Lagrange Multipliers ....................... 119
  4.3.3 Effect of the Steplength .................................... 120
  4.3.4 Convergence of the Feasibility Error ...................... 124
  4.3.5 Convergence of Primal Variables .......................... 124
  4.3.6 Implementation ............................................... 125
4.4 Asymptotically Exact Augmented Lagrangian Method ............ 126
  4.4.1 Algorithm ...................................................... 126
  4.4.2 Auxiliary Estimates ......................................... 127
  4.4.3 Convergence Analysis ....................................... 128
4.5 Adaptive Augmented Lagrangian Method ............................. 130
  4.5.1 Algorithm ...................................................... 131
  4.5.2 Convergence of Lagrange Multipliers for Large $\rho$ .... 132
  4.5.3 R-Linear Convergence for Any Initialization of $\rho$ .... 134
4.6 Semimonotonic Augmented Lagrangians (SMALE) ................. 135
  4.6.1 SMALE Algorithm ........................................... 136
  4.6.2 Relations for Augmented Lagrangians ..................... 137
  4.6.3 Convergence and Monotonicity ................................ 139
  4.6.4 Linear Convergence for Large $\rho_0$ ..................... 142
  4.6.5 Optimality of the Outer Loop ............................... 143
  4.6.6 Optimality of SMALE with Conjugate Gradients ........ 145
  4.6.7 Solution of More General Problems ....................... 147
4.7 Implementation of Inexact Augmented Lagrangians ............... 148
  4.7.1 Stopping, Modification of Constraints, 
        and Preconditioning ........................................... 148
  4.7.2 Initialization of Constants .................................. 148
4.8 Numerical Experiments ..................................................... 150
  4.8.1 Uzawa, Exact Augmented Lagrangians, and SMALE .... 150
  4.8.2 Numerical Demonstration of Optimality .................. 151
4.9 Comments and References .................................................. 152
5 **Bound Constrained Minimization** .......................................................... 155
  5.1 Review of Alternative Methods ....................................................... 157
  5.2 KKT Conditions and Related Inequalities ........................................ 158
  5.3 The Working Set Method with Exact Solutions .................................. 160
    5.3.1 Auxiliary Problems ........................................................................ 160
    5.3.2 Algorithm ......................................................................................... 161
    5.3.3 Finite Termination ............................................................................ 164
  5.4 Polyak’s Algorithm ............................................................................... 165
    5.4.1 Basic Algorithm ............................................................................... 165
    5.4.2 Finite Termination ............................................................................ 166
    5.4.3 Characteristics of Polyak’s Algorithm ........................................... 167
  5.5 Inexact Polyak’s Algorithm ................................................................. 167
    5.5.1 Looking Ahead and Estimate ........................................................... 167
    5.5.2 Looking Ahead Polyak’s Algorithm ................................................. 170
    5.5.3 Easy Re-release Polyak’s Algorithm .............................................. 171
    5.5.4 Properties of Modified Polyak’s Algorithms .................................. 172
  5.6 Gradient Projection Method ................................................................ 173
    5.6.1 Conjugate Gradient Versus Gradient Projections ............................ 174
    5.6.2 Contraction in the Euclidean Norm ................................................. 175
    5.6.3 The Fixed Steplength Gradient Projection Method ........................... 177
    5.6.4 Quadratic Functions with Identity Hessian ...................................... 178
    5.6.5 Dominating Function and Decrease of the Cost Function ................ 181
  5.7 Modified Proportioning with Gradient Projections .............................. 184
    5.7.1 MPGP Schema .................................................................................. 184
    5.7.2 Rate of Convergence ....................................................................... 186
  5.8 Modified Proportioning with Reduced Gradient Projections ............... 189
    5.8.1 MPRGP Schema .............................................................................. 189
    5.8.2 Rate of Convergence ...................................................................... 190
    5.8.3 Rate of Convergence of Projected Gradient .................................... 193
    5.8.4 Optimality ......................................................................................... 197
    5.8.5 Identification Lemma and Finite Termination .................................. 198
    5.8.6 Finite Termination for Dual Degenerate Solution ............................ 201
  5.9 Implementation of MPRGP with Optional Modifications ....................... 204
    5.9.1 Expansion Step with Feasible Half-Step ........................................ 204
    5.9.2 MPRGP Algorithm .......................................................................... 205
    5.9.3 Unfeasible MPRGP .......................................................................... 206
    5.9.4 Choice of Parameters ...................................................................... 208
    5.9.5 Dynamic Release Coefficient ........................................................... 209
  5.10 Preconditioning .................................................................................... 210
    5.10.1 Preconditioning in Face .................................................................. 210
    5.10.2 Preconditioning by Conjugate Projector ....................................... 212
  5.11 Numerical Experiments ....................................................................... 216
    5.11.1 Polyak, MPRGP, and Preconditioned MPRGP ............................... 216
    5.11.2 Numerical Demonstration of Optimality ....................................... 217
  5.12 Comments and References ................................................................... 218
6 Bound and Equality Constrained Minimization ................. 221
  6.1 Review of the Methods for Bound and Equality Constrained Problems ................. 222
  6.2 SMALBE Algorithm for Bound and Equality Constraints ...... 223
     6.2.1 KKT Conditions and Projected Gradient ................. 223
     6.2.2 SMALBE Algorithm ..................................... 223
  6.3 Inequalities Involving the Augmented Lagrangian .......... 225
  6.4 Monotonicity and Feasibility ......................................... 227
  6.5 Boundedness .................................................. 229
  6.6 Convergence .................................................. 233
  6.7 Optimality of the Outer Loop ..................................... 235
  6.8 Optimality of the Inner Loop ..................................... 237
  6.9 Solution of More General Problems ......................... 239
  6.10 Implementation .................................................. 240
  6.11 SMALBE–M .................................................. 241
  6.12 Numerical Experiments ......................................... 242
     6.12.1 Balanced Reduction of Feasibility and Gradient Errors . 242
     6.12.2 Numerical Demonstration of Optimality ................. 243
  6.13 Comments and References ........................................ 244

Part III Applications to Variational Inequalities

7 Solution of a Coercive Variational Inequality
  by FETI–DP Method ............................................. 249
  7.1 Model Coercive Variational Inequality ................. 250
  7.2 FETI–DP Domain Decomposition and Discretization ...... 251
  7.3 Optimality .................................................. 254
  7.4 Numerical Experiments ......................................... 255
  7.5 Comments and References ........................................ 256

8 Solution of a Semicoercive Variational Inequality
  by TFETI Method ............................................. 259
  8.1 Model Semicoercive Variational Inequality ................. 260
  8.2 TFETI Domain Decomposition and Discretization ...... 261
  8.3 Natural Coarse Grid ......................................... 264
  8.4 Optimality .................................................. 265
  8.5 Numerical Experiments ......................................... 267
  8.6 Comments and References ........................................ 269

References ..................................................... 271

Index .......................................................... 281
Part I

Background
1

Linear Algebra

The purpose of this chapter is to briefly review definitions, notations, and results of linear algebra that are used in the rest of our book. A few results especially developed for analysis of our algorithms are also included. There is no claim of completeness as the reader is assumed to be familiar with basic concepts of the college linear algebra such as vector spaces, linear mappings, matrix decompositions, etc. More systematic exposition and additional material can be found in the books by Strang [171], Hager [112], Demmel [31], Golub and Van Loan [103], Saad [163], and Axelsson [4]. We use without any reference basic concepts and standard results of analysis as they are reviewed in the books by Bertsekas [12] or Conn, Gould, and Toint [28].

1.1 Vectors

In this book we work with \( n \)-dimensional arithmetic vectors \( v \in \mathbb{R}^n \), where \( \mathbb{R} \) denotes the set of real numbers. The only exception is Sect. 1.8, where vectors with complex entries are considered. We denote the \( i \)th component of an arithmetic vector \( v \in \mathbb{R}^n \) by \( [v]_i \). Thus \( [v]_i = v_i \) if \( v = [v] \) is defined by its components \( v_i \). All the arithmetic vectors are considered by default to be column vectors. The relations between vectors \( u, v \in \mathbb{R}^n \) are defined componentwise. Thus \( u = v \) is equivalent to \( [u]_i = [v]_i, \ i = 1, \ldots, n \), and \( u \leq v \) is equivalent to \( [u]_i \leq [v]_i, \ i = 1, \ldots, n \). We sometimes call the elements of \( \mathbb{R}^n \) points to indicate that the concepts of length and direction are not important.

Having arithmetic vectors \( u, v \in \mathbb{R}^n \) and a scalar \( \alpha \in \mathbb{R} \), we define the \textit{addition} and \textit{multiplication by scalar} componentwise by

\[
[u + v]_i = [u]_i + [v]_i \quad \text{and} \quad [\alpha v]_i = \alpha [v]_i, \quad i = 1, \ldots, n.
\]

The rules that govern these operations, such as associativity, may be easily deduced from the related rules for computations with real numbers.
The vector analog of \( 0 \in \mathbb{R} \) is the zero vector \( \mathbf{0}_n \in \mathbb{R}^n \) with all the entries equal to zero. When the dimension can be deduced from the context, possibly using the assumption that all the expressions in our book are well defined, we often drop the subscript and write simply \( \mathbf{0} \).

A nonempty set \( V \subseteq \mathbb{R}^n \) with the operations defined above is a vector space if \( \alpha \in \mathbb{R} \) and \( u, v \in V \) imply \( u + v \in V \) and \( \alpha u \in V \). In particular, both \( \mathbb{R}^n \) and \( \{ \mathbf{0} \} \) are vector spaces. Given vectors \( v_1, \ldots, v_k \in \mathbb{R}^n \), the set

\[
\text{Span}\{v_1, \ldots, v_k\} = \{ v \in \mathbb{R}^n : v = \alpha_1 v_1 + \cdots + \alpha_k v_k, \ \alpha_i \in \mathbb{R} \}
\]

is a vector space called the linear span of \( v_1, \ldots, v_k \). If \( U \) and \( V \) are vector spaces, then the sets \( U \cap V \) and \( U + V = \{ x + y : x \in U \text{ and } y \in V \} \) are also vector spaces. If \( W = U + V \) and \( U \cap V = \{ \mathbf{0} \} \), then \( W \) is said to be the direct sum of \( U \) and \( V \). We denote it by

\[
W = U \oplus V.
\]

If \( U, V \subseteq \mathbb{R}^n \) are vector spaces and \( U \subseteq V \), then \( U \) is a subspace of \( V \).

A vector space \( V \subseteq \mathbb{R}^n \) can be spanned by different sets of vectors. A finite set of vectors \( E \subseteq \mathbb{R}^n \) that spans a given vector space \( V \neq \{ \mathbf{0} \} \) is called a basis of \( V \) if no proper subset of \( E \) spans \( V \). For example, the set of vectors

\[
S = \{ s_1, \ldots, s_n \}, \quad [s_i]_j = \delta_{ij}, \quad i, j = 1, \ldots, n,
\]

where \( \delta_{ij} \) denotes the Kronecker symbol defined by \( \delta_{ij} = 1 \) for \( i = j \) and \( \delta_{ij} = 0 \) for \( i \neq j \), is the standard basis of \( \mathbb{R}^n \). If \( E = \{ e_1, \ldots, e_d \} \) is a basis of a vector space \( V \), then \( E \) is independent, that is,

\[
\alpha_1 e_1 + \cdots + \alpha_d e_d = \mathbf{0}
\]

implies

\[
\alpha_1 = \cdots = \alpha_d = 0.
\]

Any two bases of a vector space \( V \) have the same number of vectors. We call it the dimension of \( V \) and denote it \( \dim V \). Obviously \( \dim \mathbb{R}^n = n \) and \( \dim V \leq n \) for any subspace \( V \subseteq \mathbb{R}^n \). For convenience, we define \( \dim \{ \mathbf{0} \} = 0 \).

We sometimes use the componentwise extensions of scalar functions to vectors. Thus if \( v \in \mathbb{R}^n \), then \( v^+ \) and \( v^- \) are the vectors whose \( i \)-th components are \( \max \{ [v]_i, 0 \} \) and \( \min \{ [v]_i, 0 \} \), respectively. Similarly, if \( u, v \in \mathbb{R}^n \), then \( \max \{ u, v \} \) and \( \min \{ u, v \} \) denote the vectors whose \( i \)-th components are \( \max \{ [u]_i, [v]_i \} \) and \( \min \{ [u]_i, [v]_i \} \), respectively.

If \( I \) is a nonempty subset of \( \{1, \ldots, n\} \) and \( v \in \mathbb{R}^n \), then we denote by \( [v]_I \) or simply \( v_I \) the subvector of \( v \) with components \( [v]_i, i \in I \). Thus if \( I \) has \( m \) elements, then \( v_I \in \mathbb{R}^m \), so that we can refer to the components of \( v_I \) either by the global indices \( i \in I \) or by the local indices \( j \in \{1, \ldots, m\} \). We usually rely on the reader’s judgment to recognize the appropriate type of indexing.
1.2 Matrices and Matrix Operations

Throughout the whole book, all the matrices are assumed to be real except Sect. 1.8, where also complex matrices are considered. Similarly to the related convention for vectors, the \((i, j)\)th component of a matrix \(A \in \mathbb{R}^{m \times n}\) is denoted by \([A]_{ij}\), so that \([A]_{ij} = a_{ij}\) for \(A = [a_{ij}]\) which is defined by its entries \(a_{ij}\). A matrix \(A \in \mathbb{R}^{m \times n}\) is called an \((m, n)\)-matrix; a matrix \(A \in \mathbb{R}^{n \times n}\) is called a square matrix of the order \(n\).

Having \((m, n)\)-matrices \(A, B\) and a scalar \(\alpha \in \mathbb{R}\), we define addition and multiplication by a scalar by

\[
[A + B]_{ij} = [A]_{ij} + [B]_{ij} \quad \text{and} \quad [\alpha A]_{ij} = \alpha [A]_{ij}.
\]

The rules that govern the addition of matrices and their multiplication by scalars are the same as those for corresponding vector operations.

The matrix analog of 0 is the zero matrix \(O_{mn} \in \mathbb{R}^{m \times n}\) with all the entries equal to zero. When the dimension is clear from the context, we often drop the subscripts and write simply \(O\).

Having matrices \(A \in \mathbb{R}^{m \times k}\) and \(B \in \mathbb{R}^{k \times n}\), we define their product \(AB \in \mathbb{R}^{m \times n}\) by

\[
[AB]_{ij} = \sum_{l=1}^{k} [A]_{il} [B]_{lj}.
\]

Matrix multiplication is associative, therefore we do not need to use brackets to specify the order of multiplication. In particular, given a positive integer \(k\) and a square matrix \(A\), we can define the \(k\)th power of a square matrix \(A\) by

\[
A^k = A A \ldots A.
\]

Matrix multiplication is not commutative.

The matrix counterpart of 1 \(\in \mathbb{R}\) in \(\mathbb{R}^{n \times n}\) is the identity matrix \(I_n = [\delta_{ij}]\) of the order \(n\). When the dimension may be deduced from the context, we often drop the subscripts and write simply \(I\). Thus we can write

\[
A = I A = A I
\]

for any matrix \(A\), having in mind that the order of \(I\) on the left may be different from that on the right.

Given \(A \in \mathbb{R}^{m \times n}\), we define the transposed matrix \(A^T \in \mathbb{R}^{n \times m}\) to \(A\) by \([A^T]_{ij} = [A]_{ji}\). Having matrices \(A \in \mathbb{R}^{m \times k}\) and \(B \in \mathbb{R}^{k \times n}\), it may be checked that

\[
(AB)^T = B^T A^T.
\]

A square matrix \(A\) is symmetric if \(A = A^T\).

A matrix \(A\) is positive definite if \(x^T A x > 0\) for any \(x \neq o\), positive semidefinite if \(x^T A x \geq 0\) for any \(x\), and indefinite if neither \(A\) nor \(-A\) is positive.
definite or semidefinite. We are especially interested in symmetric positive
definite (SPD) matrices.

If \( A \in \mathbb{R}^{m \times n} \), \( I \subseteq \{1, \ldots, m\} \), and \( J \subseteq \{1, \ldots, n\} \), \( I \) and \( J \) nonempty, we
denote by \( A_{IJ} \) the submatrix of \( A \) with the components \( [A]_{ij}, i \in I, j \in J \).
The local indexing of the entries of \( A_{IJ} \) is used whenever it is convenient
in a similar way as the local indexing of subvectors which was introduced in
Sect. 1.1. The full set of indices may be replaced by * so that \( A = A_{**} \) and
\( A_{I*} \) denotes the submatrix of \( A \) with the row indices belonging to \( I \).

Sometimes it is useful to rearrange the matrix operations into manip-
ulations with submatrices of given matrices called blocks. A **block matrix**
\( A \in \mathbb{R}^{m \times n} \) is defined by its blocks \( A_{ij} = A_{I_i J_j} \), where \( I_i \) and \( J_j \) denote
nonempty contiguous sets of indices decomposing \( \{1, \ldots, m\} \) and \( \{1, \ldots, n\} \),
respectively. We can use the block structure to implement matrix operations
only when the block structure of the involved matrices matches.

Very large matrices are often **sparse** in the sense that they have a small
number of nonzero entries distributed in a pattern which can be exploited to
the efficient implementation of matrix operations, to the reduction of storage
requirements, or to the effective solution of standard problems of linear alge-
bra. Such matrices arise, e.g., from the discretization of problems described
by differential operators. The matrices with a large number of nonzero entries
are often called **full** or **dense matrices**.

### 1.3 Matrices and Mappings

Each matrix \( A \in \mathbb{R}^{m \times n} \) defines the mapping which assigns to each \( x \in \mathbb{R}^n \)
the vector \( Ax \in \mathbb{R}^m \). Two important subspaces associated with this mapping
are its **range** or **image space** \( \text{Im}A \) and its **kernel** or **null space** \( \text{Ker}A \); they are
defined by

\[
\text{Im}A = \{Ax : x \in \mathbb{R}^n\} \quad \text{and} \quad \text{Ker}A = \{x \in \mathbb{R}^n : Ax = 0\}.
\]

The range of \( A \) is the span of its columns.

If \( f \) is a mapping defined on \( D \subseteq \mathbb{R}^n \) and \( \Omega \subseteq D \), then \( f|\Omega \) denotes the
**restriction** of \( f \) to \( \Omega \), that is, the mapping defined on \( \Omega \) which assigns to each
\( x \in \Omega \) the value \( f(x) \). If \( A \in \mathbb{R}^{m \times n} \) and \( \mathcal{V} \) is a subspace of \( \mathbb{R}^n \), we define \( A|\mathcal{V} \)
as a restriction of the mapping associated with \( A \) to \( \mathcal{V} \). The restriction \( A|\mathcal{V} \)
is said to be positive definite if \( x^T Ax > 0 \) for \( x \in \mathcal{V}, \ x \neq 0 \), and positive
semidefinite if \( x^T Ax \geq 0 \) for \( x \in \mathcal{V} \).

The mapping associated with \( A \) is **injective** if \( Ax = Ay \) implies \( x = y \). It
is easy to check that the mapping associated with \( A \) is injective if and only if
\( \text{Ker}A = \{0\} \). More generally, it may be proved that

\[
\dim \text{Im}A + \dim \text{Ker}A = n \quad (1.2)
\]

for any \( A \in \mathbb{R}^{m \times n} \). If \( m = n \), then \( A \) is injective if and only if \( \text{Im}A = \mathbb{R}^n \).
The rank or column rank of a matrix $A$ is equal to the dimension of the range of $A$. The column rank is known to be equal to the row rank, the number of linearly independent rows. A matrix is of full row rank or full column rank when its rank is equal to the number of its rows or columns, respectively. A matrix $A \in \mathbb{R}^{m \times n}$ is of full rank when its rank is the smaller of $m$ and $n$.

A subspace $\mathcal{V} \subseteq \mathbb{R}^n$ which satisfies

$$A \mathcal{V} = \{A x : x \in \mathcal{V}\} \subseteq \mathcal{V}$$

is an invariant subspace of $A$. Obviously

$$A(\text{Im} A) \subseteq \text{Im} A,$$

so that $\text{Im} A$ is an invariant subspace of $A$.

A projector is a square matrix $P$ that satisfies

$$P^2 = P.$$  

Such a matrix is also said to be idempotent. A vector $x \in \text{Im} P$ if and only if there is $y \in \mathbb{R}^n$ such that $x = Py$, so that

$$Px = P(Py) = Py = x.$$  

If $P$ is a projector, then $Q = I - P$ and $P^T$ are also projectors as

$$(I - P)^2 = I - 2P + P^2 = I - P \quad \text{and} \quad (P^T)^2 = (P^2)^T = P^T.$$  

Since for any $x \in \mathbb{R}^n$

$$x = Px + (I - P)x,$$

it simply follows that $\text{Im} Q = \text{Ker} P$,

$$\mathbb{R}^n = \text{Im} P + \text{Ker} P, \quad \text{and} \quad \text{Ker} P \cap \text{Im} P = \{o\}.$$  

We say that $P$ is a projector onto $\mathcal{U} = \text{Im} P$ along $\mathcal{V} = \text{Ker} P$ and $Q$ is a complementary projector onto $\mathcal{V}$ along $\mathcal{U}$. The above relations may also be rewritten as

$$\text{Im} P \oplus \text{Ker} P = \mathbb{R}^n. \quad (1.3)$$

Let $(\pi(1), \ldots, \pi(n))$ be a permutation of numbers $1, \ldots, n$. Then the mapping which assigns to each $v = [v_i] \in \mathbb{R}^n$ a vector $[v_{\pi(1)}, \ldots, v_{\pi(n)}]^T$ is associated with the permutation matrix

$$P = [s_{\pi(1)}, \ldots, s_{\pi(n)}],$$

where $s_i$ denotes the $i$th column of the identity matrix $I_n$. If $P$ is a permutation matrix, then

$$PP^T = P^TP = I.$$  

Notice that if $B$ is a matrix obtained from a matrix $A$ by reordering of the rows of $A$, then there is a permutation matrix $P$ such that $B = PA$. Similarly, if $B$ is a matrix obtained from $A$ by reordering of the columns of $A$, then there is a permutation matrix $P$ such that $B = AP$. 

1.4 Inverse and Generalized Inverse Matrices

If $A$ is a square full rank matrix, then there is the unique inverse matrix $A^{-1}$ such that

$$AA^{-1} = A^{-1}A = I.$$  \hfill (1.4)

The mapping associated with $A^{-1}$ is inverse to that associated with $A$.

If $A^{-1}$ exists, we say that $A$ is nonsingular. A square matrix is singular if its inverse matrix does not exist. Any positive definite matrix is nonsingular.

If $P$ is a permutation matrix, then $P$ is nonsingular and $P^{-1} = P^T$.

If $A$ is a nonsingular matrix, then $A^{-1}b$ is the unique solution of the system of linear equations $Ax = b$.

If $A$ is a nonsingular matrix, then we can transpose (1.4) and use (1.1) to get

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

so that

$$(A^T)^{-1} = (A^{-1})^T.$$  \hfill (1.5)

It follows that if $A$ is symmetric, then $A^{-1}$ is symmetric.

If $A \in \mathbb{R}^{n \times n}$ is positive definite, then $A^{-1}$ is also positive definite, as any vector $x \neq o$ can be expressed as $x = Ay$, $y \neq o$, and

$$x^T A^{-1} x = (Ay)^T A^{-1} Ay = y^T A^T y = y^T Ay > 0.$$

If $A$ and $B$ are nonsingular matrices, then it is easy to check that $AB$ is also nonsingular and

$$(AB)^{-1} = B^{-1}A^{-1}.$$  

If $U, V \in \mathbb{R}^{m \times n}$, $m < n$, and $A$, $A + UTV$ are nonsingular, then it can be verified directly that

$$(A + UTV)^{-1} = A^{-1} - A^{-1}U^T(I + VA^{-1}U^T)^{-1}VA^{-1}.$$  \hfill (1.6)

The formula (1.6) is known as Sherman–Morrison–Woodbury’s formula (see [103, p. 51]). The formula is useful in theory and for evaluation of the inverse matrix to a low rank perturbation of $A$ provided $A^{-1}$ is known.

If $B \in \mathbb{R}^{m \times n}$ denotes a full row rank matrix, $A \in \mathbb{R}^{n \times n}$ is positive definite, and $y \neq o$, then $z = B^Ty \neq o$ and

$$y^TBA^{-1}B^Ty = z^TA^{-1}z > 0.$$

Thus if $A$ is positive definite and $B$ is a full row rank matrix such that $BA^{-1}B^T$ is well defined, then the latter matrix is also positive definite.

A real matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ is called a (nonsingular) $M$-matrix if $a_{ij} \leq 0$ for $i \neq j$ and if all entries of $A^{-1}$ are nonnegative. If
1.5 Direct Methods for Solving Linear Equations

\[ a_{ii} > \sum_{j \neq i}^{n} |a_{ij}|, \quad i = 1, \ldots, n, \]

then \( A \) is an \( M \)-matrix (see Fiedler and Pták [88] or Axelsson [4, Chap. 6]).

If \( A \in \mathbb{R}^{m \times n} \) and \( b \in \text{Im} A \), then we can express a solution of the system of linear equations \( Ax = b \) by means of a left generalized inverse matrix \( A^+ \in \mathbb{R}^{n \times m} \) which satisfies \( AA^+ A = A \). Indeed, if \( b \in \text{Im} A \), then there is \( y \) such that \( b = Ay \) and \( x = A^+ b \) satisfies

\[ A\bar{x} = AA^+ b = AA^+ Ay = Ay = b. \]

Thus \( A^+ \) acts on the range of \( A \) like the inverse matrix. If \( A \) is a nonsingular square matrix, then obviously

\[ A^+ = A^{-1}. \]

Moreover, if \( S \in \mathbb{R}^{n \times p} \) is such that \( AS = O \) and \( N \in \mathbb{R}^{n \times p} \), then \( (A^+) + SN^T \) is also a left generalized inverse as

\[ A \left( (A^+) + SN^T \right) A = AA^+ A + ASN^T A = A. \]

If \( A \) is a symmetric singular matrix, then there is a permutation matrix \( P \) such that

\[ A = P^T \begin{bmatrix} B & C^T \\ C & CB^{-1}C^T \end{bmatrix} P, \]

where \( B \) is a nonsingular matrix whose dimension is equal to the rank of \( A \). It may be verified directly that the matrix

\[ A^\# = P^T \begin{bmatrix} B^{-1} O^T \\ O & O \end{bmatrix} P \quad (1.7) \]

is a left generalized inverse of \( A \). If \( A \) is symmetric positive semidefinite, then \( A^\# \) is also symmetric positive semidefinite. Notice that if \( AS = O \), then \( A^+ = A^\# + SS^T \) is also a symmetric positive semidefinite generalized inverse.

1.5 Direct Methods for Solving Linear Equations

The inverse matrix is a useful tool for theoretical developments, but not for computations, especially when sparse matrices are involved. The reason is that the inverse matrix is usually full, so that its evaluation results in large storage requirements and high computational costs. It is often much more efficient to implement the multiplication of a vector by the inverse matrix by solving the related system of linear equations. We recall here briefly the direct methods, which reduce solving of the original system of linear equations to solving of a system or systems of linear equations with triangular matrices.
A matrix $L = [l_{ij}]$ is lower triangular if $l_{ij} = 0$ for $i < j$. It is easy to solve a system $Lx = b$ with the nonsingular lower triangular matrix $L \in \mathbb{R}^n$. As there is only one unknown in the first equation, we can find it and then substitute it into the remaining equations to obtain a system with the same structure, but with only $n - 1$ remaining unknowns. We can repeat the procedure until we find all the components of $x$.

A similar procedure, but starting from the last equation, can be applied to a system with the nonsingular upper triangular matrix $U = [u_{ij}]$ with $u_{ij} = 0$ for $i > j$.

The solution costs of a system with triangular matrices is proportional to the number of its nonzero entries. In particular, the solution of a system of linear equations with a diagonal matrix $D = [d_{ij}]$, $d_{ij} = 0$ for $i \neq j$, reduces to the solution of a sequence of linear equations with one unknown.

If we are to solve the system of linear equations with a nonsingular matrix, we can use systematically equivalent transformations that do not change the solution in order to modify the original system to that with an upper triangular matrix. It is well-known that the solutions of a system of linear equations are the same as the solutions of a system of linear equations obtained from the original system by interchanging two equations, replacing an equation by its nonzero multiple, or adding a multiple of one equation to another equation. The Gauss elimination for the solution of a system of linear equations with a nonsingular matrix thus consists of two steps: the forward reduction, which exploits equivalent transformations to reduce the original system to the system with an upper triangular matrix, and the backward substitution, which solves the resulting system with the upper triangular matrix.

Alternatively, we can use suitable matrix factorizations. For example, it is well-known that any positive definite matrix $A$ can be decomposed into the product

$$A = LL^T,$$

where $L$ is a nonsingular lower triangular matrix with positive diagonal entries. Having the decomposition, we can evaluate $z = A^{-1}x$ by solving the systems

$$Ly = x \quad \text{and} \quad L^Tz = y.$$  

The factorization-based solvers may be especially useful when we are to solve several systems of equations with the same coefficients but different right-hand sides coming one after another.

The method of evaluation of the factor $L$ is known as the Cholesky factorization. The Cholesky factor $L$ can be computed in a number of equivalent ways. For example, we may compute it column by column. Suppose that

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

Substituting for $A$ and $L$ into (1.8) and comparing the corresponding terms immediately reveals that
1.5 Direct Methods for Solving Linear Equations

\[ l_{11} = \sqrt{a_{11}}, \quad l_1 = l_{11}^{-1} a_1, \quad L_{22} L_{22}^T = A_{22} - l_1 l_1^T. \] (1.9)

This gives us the first column of \( L \), and the remaining factor \( L_{22} \) is simply the Cholesky factor of the Schur complement \( A_{22} - l_1 l_1^T \) which is known to be positive definite, so we can find its first column by the above procedure. The algorithm can be implemented to exploit a sparsity pattern of \( A \), e.g., when \( A \in \mathbb{R}^{n \times n} \) is a band matrix with \( a_{ij} = 0 \) for \( |i - j| > b \), \( b \ll n \).

If \( A \in \mathbb{R}^{n \times n} \) is only positive semidefinite, it can happen that \( a_{11} = 0 \). Then

\[ 0 \leq x^T A x = y^T A_{22} y + 2 x_1 a_1^T y \]

for any vector \( x = [x_1, y^T]^T \). The inequality implies that \( a_1 = 0 \), as otherwise we could take \( y = -a_1 \) and large \( x_1 \) to get

\[ y^T A_{22} y + 2 x_1 a_1^T y = a_1^T A_{22} a_1 - 2 x_1 \| a_1 \|^2 < 0. \]

Thus for \( A \) symmetric positive semidefinite and \( a_{11} = 0 \), (1.9) reduces to

\[ l_{11} = 0, \quad l_1 = 0, \quad L_{22} L_{22}^T = A_{22}. \] (1.10)

Of course, this simple modification assumes exact arithmetics. In the computer arithmetics, the decision whether \( a_{11} \) is to be treated as zero depends on some small \( \varepsilon > 0 \).

In some important applications, it is possible to exploit additional information. In mechanics, e.g., the basis of the kernel of the stiffness matrix of a floating body is formed by three (2D) or six (3D) known and independent rigid body motions. Any basis of the kernel of a matrix can be used to identify the zero rows (and columns) of a Cholesky factor by means of the following lemma.

**Lemma 1.1.** Let \( A = LL^T \) denote a triangular decomposition of a symmetric positive semidefinite matrix \( A \), let \( A e = 0 \), and let \( l(e) \) denote the largest index of a nonzero entry of \( e \in \text{Ker} A \), so that

\[ [e]_{l(e)} \neq 0 \quad \text{and} \quad [e]_j = 0 \quad \text{for} \quad j > l(e). \]

Then

\[ [L]_{l(e)l(e)} = 0. \]

**Proof.** If \( A e = 0 \), then

\[ e^T A e = e^T L L^T e = (L^T e)^T (L^T e) = 0. \]

Thus \( L^T e = 0 \) and in particular

\[ [L^T e]_{l(e)} = [L]_{l(e)l(e)} [e]_{l(e)} = 0. \]

Since \( [e]_{l(e)} \neq 0 \), we have \( [L]_{l(e)l(e)} = 0 \). \( \square \)
Let $A \in \mathbb{R}^{n \times n}$ be positive semidefinite and let $R \in \mathbb{R}^{n \times d}$ denote a full column rank matrix such that $\text{Ker} A = \text{Im} R$. Observing that application of equivalent transformations to the columns of $R$ preserves the image space and the rank of $R$, we can modify the forward reduction to find $R$ which satisfies

$$l(R_{*1}) < \cdots < l(R_{*d}).$$

The procedure can be described by the following transformations of $R$: transpose $R$, reverse the order of columns, apply the forward reduction, reverse the order of columns back, and transpose the resulting matrix back. Then $l(R_{*1}), \ldots, l(R_{*d})$ are by Lemma 1.1 the indices of zero columns of a factor of the modified Cholesky factorization; the factor cannot have any other zero columns due to the rank argument. The procedure has been described and tested in Menšík [151]. Denoting by the crosses and dots the nonzero and undetermined entries, respectively, the relations between the pivots of $R$ and the zero columns of the Cholesky factor $L$ can be illustrated by

$$R = \begin{bmatrix}
\ldots \\
\ldots \\
\times \\
0 \\
0 \times
\end{bmatrix} \Rightarrow L = \begin{bmatrix}
\times & 0 & 0 & 0 \\
. & \times & 0 & 0 \\
. & . & 0 & 0 \\
. & . & 0 & x \\
. & . & . & 0 \\
\end{bmatrix}.$$

Alternatively, we can combine the basic algorithm with a suitable rank revealing decomposition, such as the singular value decomposition (SVD) introduced in Sect. 1.9. For example, Frahat and Gérardin [82] proposed to start with the Cholesky decomposition and to switch to SVD in case of doubts.

### 1.6 Norms

General concepts of size and distance in a vector space are expressed by norms. A norm on $\mathbb{R}^n$ is a function which assigns to each $x \in \mathbb{R}^n$ a number $\|x\| \in \mathbb{R}$ in such a way that for any vectors $x, y \in \mathbb{R}^n$ and any scalar $\alpha \in \mathbb{R}$, the following three conditions are satisfied:

(i) $\|x\| \geq 0$, and $\|x\| = 0$ if and only if $x = 0$.

(ii) $\|x + y\| \leq \|x\| + \|y\|$.

(iii) $\|\alpha x\| = |\alpha| \|x\|$.

It is easy to check that the functions

$$\|x\|_1 = |x_1| + \cdots + |x_n| \quad \text{and} \quad \|x\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$$

are norms. They are called $\ell_1$ and $\ell_\infty$ norms, respectively. We often use the Euclidean norm defined by

$$\|x\|_2 = \sqrt{x_1^2 + \cdots + x_n^2}.$$
The norms on $\mathbb{R}^n$ introduced above satisfy the inequalities

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

Given a norm defined on the domain and the range of a matrix $A$, we can define the \textit{induced norm} $\|A\|$ of $A$ by

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

If $B \neq 0$, then

$$\|AB\| = \sup_{x \neq 0} \frac{\|ABx\|}{\|x\|} = \sup_{Bx \neq 0} \frac{\|ABx\|}{\|Bx\|} \|Bx\| \leq \sup_{y \in \text{Im}B} \frac{\|Ay\|}{\|y\|} \|Bx\| \|B\|.$$ 

It follows easily that the induced norm is \textit{submultiplicative}, i.e.,

$$\|AB\| \leq \|A\| \|\text{Im}B\| \|B\| \leq \|A\| \|B\|. \quad (1.11)$$

If $A = [a_{ij}] \in \mathbb{R}^{m \times n}$ and $x = [x_i] \in \mathbb{R}^n$, then

$$\|Ax\|_\infty = \max_{i=1,\ldots,m} \left| \sum_{j=1}^n a_{ij} x_j \right| \leq \max_{i=1,\ldots,m} \sum_{j=1}^n |a_{ij}| \|x\|_\infty \leq \max_{j=1,\ldots,n} \sum_{i=1}^m |a_{ij}|,$$

that is, $\|A\|_\infty \leq \max_{i=1,\ldots,m} \sum_{j=1}^n |a_{ij}|$. Since the last inequality turns into the equality for a vector $x$ with suitably chosen entries $x_i \in \{1, -1\}$, we have

$$\|A\|_\infty = \max_{i=1,\ldots,m} \sum_{j=1}^n |a_{ij}|. \quad (1.12)$$

Similarly

$$\|Ax\|_1 = \sum_{i=1}^m \left| \sum_{j=1}^n a_{ij} x_j \right| \leq \sum_{j=1}^n |x_j| \sum_{i=1}^m |a_{ij}| \leq \|x\|_1 \max_{j=1,\ldots,n} \sum_{i=1}^m |a_{ij}|,$$

that is, $\|A\|_1 \leq \max_{i=1,\ldots,n} \sum_{j=1}^m |a_{ij}|$. Taking for the vector $x$ a suitably chosen column of the identity matrix $I_n$, we get

$$\|A\|_1 = \max_{j=1,\ldots,n} \sum_{i=1}^m |a_{ij}| = \|A^T\|_\infty. \quad (1.13)$$

The matrix norms induced by $\ell_1$ and $\ell_\infty$ norms are relatively inexpensive to compute. If $A \in \mathbb{R}^{m \times n}$, they may be used to estimate the typically expensive Euclidean norm $\|A\|_2$ by means of the inequalities

$$\|A\|_\infty \leq \sqrt{n} \|A\|_2 \leq n \|A\|_1 \leq n \sqrt{m} \|A\|_2 \leq nm \|A\|_\infty.$$ 

Another useful inequality is

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}. \quad (1.14)$$
1.7 Scalar Products

General concepts of length and angle in a vector space are introduced by means of a scalar product; it is the mapping which assigns to each couple \( x, y \in \mathbb{R}^n \) a number \( \langle x, y \rangle \in \mathbb{R} \) in such a way that for any vectors \( x, y, z \in \mathbb{R}^n \) and any scalar \( \alpha \in \mathbb{R} \), the following four conditions are satisfied:

(i) \( \langle x, y + z \rangle = \langle x, y \rangle + \langle x, z \rangle \).
(ii) \( \langle \alpha x, y \rangle = \alpha \langle x, y \rangle \).
(iii) \( \langle x, y \rangle = \langle y, x \rangle \).
(iv) \( \langle x, x \rangle > 0 \) for \( x \neq o \).

We often use the Euclidean scalar product or Euclidean inner product which assigns to each couple of vectors \( x, y \in \mathbb{R}^n \) a number defined by

\[
(x, y) = x^T y.
\]

If \( A \) is a symmetric positive definite matrix, then we can define the more general \( A \)-scalar product on \( \mathbb{R}^n \) by

\[
(x, y)_A = x^T Ay.
\]

Using a scalar product, we can define the norm \( \|x\| \) of \( x \) and the angle \( \alpha \) between \( x \) and \( y \) by

\[
\|x\|^2 = \langle x, x \rangle, \quad \cos \alpha = \frac{\langle x, y \rangle}{\|x\|\|y\|}.
\]

We denote for any \( x \in \mathbb{R}^n \) its Euclidean norm \( \|x\| \) and \( A \)-norm by

\[
\|x\| = (\langle x, x \rangle)^{1/2}, \quad \|x\|_A = (\langle x, x \rangle_A)^{1/2}.
\]

It is easy to see that any norm induced by a scalar product satisfies the properties (i) and (iii) of the norm. The property (ii) follows from the Cauchy–Schwarz inequality

\[
\langle x, y \rangle^2 \leq \|x\| \|y\|, \tag{1.15}
\]

which is valid for any \( x, y \in \mathbb{R}^n \) and any scalar product. The bound is tight in the sense that the inequality becomes the equality when \( x, y \) are dependent. The property (ii) of the norm then follows by

\[
\|x + y\|^2 = \|x\|^2 + 2\langle x, y \rangle + \|y\|^2 \leq \|x\|^2 + 2\|x\|\|y\| + \|y\|^2 = (\|x\| + \|y\|)^2.
\]

A pair of vectors \( x \) and \( y \) is orthogonal (with respect to a given scalar product) if

\[
\langle x, y \rangle = 0.
\]

If the scalar product is not specified, then we assume by default the Euclidean scalar product. The vectors \( x \) and \( y \) that are orthogonal in \( A \)-scalar product are also called \( A \)-conjugate or briefly conjugate.
Two sets of vectors $\mathcal{E}$ and $\mathcal{F}$ are orthogonal (also stated “$\mathcal{E}$ orthogonal to $\mathcal{F}$”) if every $x \in \mathcal{E}$ is orthogonal to any $y \in \mathcal{F}$. The set $\mathcal{E}^\perp$ of all the vectors of $\mathbb{R}^n$ that are orthogonal to $\mathcal{E} \subseteq \mathbb{R}^n$ is a vector space called the orthogonal complement of $\mathcal{E}$. If $\mathcal{E} \subseteq \mathbb{R}^n$, then
\[ \mathbb{R}^n = \text{Span} \mathcal{E} \oplus \mathcal{E}^\perp. \]

A set of vectors $\mathcal{E}$ is orthogonal if its elements are pairwise orthogonal, i.e., any $x \in \mathcal{E}$ is orthogonal to any $y \in \mathcal{E}$, $y \neq x$. A set of vectors $\mathcal{E}$ is orthonormal if it is orthogonal and $\langle x, x \rangle = 1$ for any $x \in \mathcal{E}$.

Any orthogonal set $\mathcal{E} = \{e_1, \ldots, e_n\}$ of nonzero vectors $e_i$ is independent. Indeed, if
\[ \alpha_1 e_1 + \cdots + \alpha_n e_n = 0, \]
then we can take the scalar product of both sides of the equation with $e_i$ and use the assumption on orthogonality of $\mathcal{E}$ to get that
\[ \alpha_i \langle e_i, e_i \rangle = 0, \]
so that $\alpha_i = 0$.

If $\mathcal{E}$ is an orthonormal basis of a vector space $\mathcal{V} \subseteq \mathbb{R}^n$, then the same procedure as above may be used to get conveniently the coordinates $\xi_i$ of any $x \in \mathcal{V}$. For example, if $\mathcal{E}$ is orthonormal with respect to the Euclidean scalar product, it is enough to multiply $x = \xi_1 e_1 + \cdots + \xi_n e_n$ on the left by $e_i^T$ to get
\[ \xi_i = e_i^T x. \]

A square matrix $U$ is orthogonal if $U^T U = I$, that is, $U^{-1} = U^T$. Multiplication by an orthogonal matrix $U$ preserves both the angles between any two vectors and the Euclidean norm of any vector as
\[ (Ux)^T U y = x^T U^T U y = x^T y. \]

A matrix $P \in \mathbb{R}^{n \times n}$ is an orthogonal projector if $P$ is a projector, i.e., $P^2 = P$, and $\text{Im} P$ is orthogonal to $\text{Ker} P$. The latter condition can be rewritten equivalently as
\[ P^T (I - P) = O. \]
It simply follows that
\[ P^T = P^T P = P, \]
so that orthogonal projectors are symmetric matrices and symmetric projectors are orthogonal projectors. If $P$ is an orthogonal projector, then $I - P$ is also an orthogonal projector as
\[ (I - P)^2 = I - 2P + P^2 = I - P \quad \text{and} \quad (I - P)^T P = (I - P) P = O. \]