Wiley Series in Discrete Mathematics and Optimization

AN INTRODUCTION TO OPTIMIZATION

FOURTH EDITION

Edwin K. P. Chong
Stanislaw H. Żak

WILEY

WWW. LINK AVAILABLE

AN INTRODUCTION TO OPTIMIZATION

WILEY SERIES IN
DISCRETE MATHEMATICS AND OPTIMIZATION

A complete list of titles in this series appears at the end of this volume.

AN INTRODUCTION TO OPTIMIZATION

Fourth Edition

Edwin K. P. Chong
Colorado State University

Stanislaw H. Żak
Purdue University

WILEY

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey

Published simultaneously in Canada

Library of Congress Cataloging-in-Publication Data

Chong, Edwin Kah Pin.
An introduction to optimization / Edwin K. P. Chong, Colorado State University, Stanislaw H. Zak, Purdue University. — Fourth edition.
pages cm
Summary: "The purpose of the book is to give the reader a working knowledge of optimization theory and methods" — Provided by publisher.
Includes bibliographical references and index.
ISBN 978-1-118-27901-4 (hardback)
1. Mathematical optimization. I. Zak, Stanislaw H. II. Title.
QA402.5.C476 2012
519.6—dc23 2012031772

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

To my wife, Yat-Yee,
and to my parents, Paul
and Julienne Chong.
Edwin K. P. Chong

To JMJ; my wife,
Mary Ann; and my
parents, Janina and
Konstanty Żak.
Stanislaw H. Żak

CONTENTS

PREFACE

Optimization is central to any problem involving decision making, whether in engineering or in economics. The task of decision making entails choosing among various alternatives. This choice is governed by our desire to make the "best" decision. The measure of goodness of the alternatives is described by an objective function or performance index. Optimization theory and methods deal with selecting the best alternative in the sense of the given objective function.

The area of optimization has received enormous attention in recent years, primarily because of the rapid progress in computer technology, including the development and availability of user-friendly software, high-speed and parallel processors, and artificial neural networks. A clear example of this phenomenon is the wide accessibility of optimization software tools such as the Optimization Toolbox of MATLAB[1] and the many other commercial software packages.

There are currently several excellent graduate textbooks on optimization theory and methods (e.g., [3], [39], [43], [51], [87], [88], [104], [129]), as well as undergraduate textbooks on the subject with an emphasis on engineering design (e.g., [1] and [109]). However, there is a need for an introductory

[1]MATLAB is a registered trademark of The MathWorks, Inc.

textbook on optimization theory and methods at a senior undergraduate or beginning graduate level. The present text was written with this goal in mind. The material is an outgrowth of our lecture notes for a one-semester course in optimization methods for seniors and beginning graduate students at Purdue University, West Lafayette, Indiana. In our presentation, we assume a working knowledge of basic linear algebra and multivariable calculus. For the reader's convenience, a part of this book (Part I) is devoted to a review of the required mathematical background material. Numerous figures throughout the text complement the written presentation of the material. We also include a variety of exercises at the end of each chapter. A solutions manual with complete solutions to the exercises is available from the publisher to instructors who adopt this text. Some of the exercises require using MATLAB. The student edition of MATLAB is sufficient for all of the MATLAB exercises included in the text. The MATLAB source listings for the MATLAB exercises are also included in the solutions manual.

The purpose of the book is to give the reader a working knowledge of optimization theory and methods. To accomplish this goal, we include many examples that illustrate the theory and algorithms discussed in the text. However, it is not our intention to provide a cookbook of the most recent numerical techniques for optimization; rather, our goal is to equip the reader with sufficient background for further study of advanced topics in optimization.

The field of optimization is still a very active research area. In recent years, various new approaches to optimization have been proposed. In this text, we have tried to reflect at least some of the flavor of recent activity in the area. For example, we include a discussion of randomized search methods—these include particle swarm optimization and genetic algorithms, topics of increasing importance in the study of complex adaptive systems. There has also been a recent surge of applications of optimization methods to a variety of new problems. An example of this is the use of descent algorithms for the training of feedforward neural networks. An entire chapter in the book is devoted to this topic. The area of neural networks is an active area of ongoing research, and many books have been devoted to this subject. The topic of neural network training fits perfectly into the framework of unconstrained optimization methods. Therefore, the chapter on feedforward neural networks not only provides an example of application of unconstrained optimization methods but also gives the reader an accessible introduction to what is currently a topic of wide interest.

The material in this book is organized into four parts. Part I contains a review of some basic definitions, notations, and relations from linear algebra, geometry, and calculus that we use frequently throughout the book. In Part II we consider unconstrained optimization problems. We first discuss some theoretical foundations of set-constrained and unconstrained optimization, including necessary and sufficient conditions for minimizers and maximizers. This is followed by a treatment of various iterative optimization algorithms, including line search methods, together with their properties. A discussion of global search algorithms is included in this part. We also analyze the least-squares optimization problem and the associated recursive least-squares algorithm. Parts III and IV are devoted to constrained optimization. Part III deals with linear programming problems, which form an important class of constrained optimization problems. We give examples and analyze properties of linear programs, and then discuss the simplex method for solving linear programs. We also provide a brief treatment of dual linear programming problems. We then describe some nonsimplex algorithms for solving linear programs: Khachiyan's method, the affine scaling method, and Karmarkar's method. We wrap up Part III by discussing integer linear programming problems. In Part IV we treat nonlinear constrained optimization. Here, as in Part II, we first present some theoretical foundations of nonlinear constrained optimization problems, including convex optimization problems. We then discuss different algorithms for solving constrained optimization problems. We also treat multiobjective optimization.

Although we have made every effort to ensure an error-free text, we suspect that some errors remain undetected. For this purpose, we provide online updated errata that can be found at the Web site for the book, accessible via

`http://www.wiley.com/mathematics`

We are grateful to several people for their help during the course of writing this book. In particular, we thank Dennis Goodman of Lawrence Livermore Laboratories for his comments on early versions of Part II and for making available to us his lecture notes on nonlinear optimization. We thank Moshe Kam of Drexel University for pointing out some useful references on nonsimplex methods. We are grateful to Ed Silverman and Russell Quong for their valuable remarks on Part I of the first edition. We also thank the students of ECE 580 at Purdue University and ECE 520 and MATH 520 at Colorado State University for their many helpful comments and suggestions. In particular, we are grateful to Christopher Taylor for his diligent proofreading of early manuscripts of this book. This fourth edition incorporates many valuable suggestions of users of the first, second, and third editions, to whom we are grateful.

E. K. P. CHONG AND S. H. ŻAK
Fort Collins, Colorado, and West Lafayette, Indiana

PART I

MATHEMATICAL REVIEW

CHAPTER 1

METHODS OF PROOF AND SOME NOTATION

1.1 Methods of Proof

Consider two statements, "A" and "B," which could be either true or false. For example, let "A" be the statement "John is an engineering student," and let "B" be the statement "John is taking a course on optimization." We can combine these statements to form other statements, such as "A and B" or "A or B." In our example, "A and B" means "John is an engineering student, and he is taking a course on optimization." We can also form statements such as "not A," "not B," "not (A and B)," and so on. For example, "not A" means "John is not an engineering student." The truth or falsity of the combined statements depend on the truth or falsity of the original statements, "A" and "B." This relationship is expressed by means of truth tables; see Tables 1.1 and 1.2.

From the tables, it is easy to see that the statement "not (A and B)" is equivalent to "(not A) or (not B)" (see Exercise 1.3). This is called *DeMorgan's law.*

In proving statements, it is convenient to express a combined statement by a *conditional*, such as "A implies B," which we denote "A$\Rightarrow$B." The conditional

Table 1.1 Truth Table for "A and B" and "A or B"

A	B	A and B	A or B
F	F	F	F
F	T	F	T
T	F	F	T
T	T	T	T

Table 1.2 Truth Table for "not A"

A	not A
F	T
T	F

Table 1.3 Truth Table for Conditionals and Biconditionals

A	B	A⇒B	A ⇐ B	A ⇔ B
F	F	T	T	T
F	T	T	F	F
T	F	F	T	F
T	T	T	T	T

"A⇒B" is simply the combined statement "(not A) or B" and is often also read "A only if B," or "if A then B," or "A is sufficient for B," or "B is necessary for A."

We can combine two conditional statements to form a *biconditional* statement of the form "A⇔B," which simply means "(A⇒B) and (B⇒A)." The statement "A⇔B" reads "A if and only if B," or "A is equivalent to B," or "A is necessary and sufficient for B." Truth tables for conditional and biconditional statements are given in Table 1.3.

It is easy to verify, using the truth table, that the statement "A⇒B" is equivalent to the statement "(not B)⇒(not A)." The latter is called the *contrapositive* of the former. If we take the contrapositive to DeMorgan's law, we obtain the assertion that "not (A or B)" is equivalent to "(not A) and (not B)."

Most statements we deal with have the form "A⇒B." To prove such a statement, we may use one of the following three different techniques:

1. The direct method

2. Proof by contraposition

3. Proof by contradiction or *reductio ad absurdum*

In the case of the *direct method*, we start with "A," then deduce a chain of various consequences to end with "B."

A useful method for proving statements is *proof by contraposition*, based on the equivalence of the statements "A⇒B" and "(not B)⇒(not A)." We start with "not B," then deduce various consequences to end with "not A" as a conclusion.

Another method of proof that we use is *proof by contradiction*, based on the equivalence of the statements "A⇒B" and "not (A and (not B))." Here we begin with "A and (not B)" and derive a contradiction.

Occasionally, we use the *principle of induction* to prove statements. This principle may be stated as follows. Assume that a given property of positive integers satisfies the following conditions:

- The number 1 possesses this property.
- If the number n possesses this property, then the number $n+1$ possesses it too.

The principle of induction states that under these assumptions any positive integer possesses the property.

The principle of induction is easily understood using the following intuitive argument. If the number 1 possesses the given property, then the second condition implies that the number 2 possesses the property. But, then again, the second condition implies that the number 3 possesses this property, and so on. The principle of induction is a formal statement of this intuitive reasoning.

For a detailed treatment of different methods of proof, see [130].

1.2 Notation

Throughout, we use the following notation. If X is a set, then we write $x \in X$ to mean that x is an element of X. When an object x is not an element of a set X, we write $x \notin X$. We also use the "curly bracket notation" for sets, writing down the first few elements of a set followed by three dots. For example, $\{x_1, x_2, x_3, \ldots\}$ is the set containing the elements x_1, x_2, x_3, and so on. Alternatively, we can explicitly display the law of formation. For example, $\{x : x \in \mathbb{R},\ x > 5\}$ reads "the set of all x such that x is real and x is greater than 5." The colon following x reads "such that." An alternative notation for the same set is $\{x \in \mathbb{R} : x > 5\}$.

If X and Y are sets, then we write $X \subset Y$ to mean that every element of X is also an element of Y. In this case, we say that X is a *subset* of Y. If X and Y are sets, then we denote by $X \setminus Y$ ("X minus Y") the set of all points in X that are not in Y. Note that $X \setminus Y$ is a subset of X. The notation $f : X \to Y$ means "f is a function from the set X into the set Y." The symbol := denotes arithmetic assignment. Thus, a statement of the form $x := y$ means "x becomes y." The symbol $\triangleq$ means "equals by definition."

Throughout the text, we mark the end of theorems, lemmas, propositions, and corollaries using the symbol □. We mark the end of proofs, definitions, and examples by ■.

We use the IEEE style when citing reference items. For example, [77] represents reference number 77 in the list of references at the end of the book.

EXERCISES

1.1 Construct the truth table for the statement "(not B)⇒(not A)," and use it to show that this statement is equivalent to the statement "A⇒B."

1.2 Construct the truth table for the statement "not (A and (not B))," and use it to show that this statement is equivalent to the statement "A⇒B."

1.3 Prove DeMorgan's law by constructing the appropriate truth tables.

1.4 Prove that for any statements A and B, we have "A ⇔ (A and B) or (A and (not B))." This is useful because it allows us to prove a statement A by proving the two separate cases "(A and B)" and "(A and (not B))." For example, to prove that $|x| \geq x$ for any $x \in \mathbb{R}$, we separately prove the cases "$|x| \geq x$ and $x \geq 0$" and "$|x| \geq x$ and $x < 0$." Proving the two cases turns out to be easier than proving the statement $|x| \geq x$ directly (see Section 2.4 and Exercise 2.7).

1.5 (This exercise is adopted from [22, pp. 80–81]) Suppose that you are shown four cards, laid out in a row. Each card has a letter on one side and a number on the other. On the visible side of the cards are printed the symbols

$$S \quad 8 \quad 3 \quad A$$

Determine which cards you should turn over to decide if the following rule is true or false: "If there is a vowel on one side of the card, then there is an even number on the other side."

CHAPTER 2

VECTOR SPACES AND MATRICES

2.1 Vector and Matrix

We define a *column n-vector* to be an array of n numbers, denoted

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}.$$

The number a_i is called the ith component of the vector $\boldsymbol{a}$. Denote by $\mathbb{R}$ the set of real numbers and by $\mathbb{R}^n$ the set of column n-vectors with real components. We call $\mathbb{R}^n$ an n-dimensional *real vector space*. We commonly denote elements of $\mathbb{R}^n$ by lowercase bold letters (e.g., $\boldsymbol{x}$). The components of $\boldsymbol{x} \in \mathbb{R}^n$ are denoted $x_1, \ldots, x_n$.

We define a *row n-vector* as

$$[a_1, a_2, \ldots, a_n].$$

An Introduction to Optimization, Fourth Edition.

The *transpose* of a given column vector $\boldsymbol{a}$ is a row vector with corresponding elements, denoted $\boldsymbol{a}^\top$. For example, if

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix},$$

then

$$\boldsymbol{a}^\top = [a_1, a_2, \ldots, a_n].$$

Equivalently, we may write $\boldsymbol{a} = [a_1, a_2, \ldots, a_n]^\top$. Throughout the text we adopt the convention that the term *vector* (without the qualifier *row* or *column*) refers to a column vector.

Two vectors $\boldsymbol{a} = [a_1, a_2, \ldots, a_n]^\top$ and $\boldsymbol{b} = [b_1, b_2, \ldots, b_n]^\top$ are equal if $a_i = b_i$, $i = 1, 2, \ldots, n$.

The sum of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, denoted $\boldsymbol{a} + \boldsymbol{b}$, is the vector

$$\boldsymbol{a} + \boldsymbol{b} = [a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n]^\top.$$

The operation of addition of vectors has the following properties:

1. The operation is commutative:

$$\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{b} + \boldsymbol{a}.$$

2. The operation is associative:

$$(\boldsymbol{a} + \boldsymbol{b}) + \boldsymbol{c} = \boldsymbol{a} + (\boldsymbol{b} + \boldsymbol{c}).$$

3. There is a zero vector

$$\boldsymbol{0} = [0, 0, \ldots, 0]^\top$$

such that

$$\boldsymbol{a} + \boldsymbol{0} = \boldsymbol{0} + \boldsymbol{a} = \boldsymbol{a}.$$

The vector

$$[a_1 - b_1, a_2 - b_2, \ldots, a_n - b_n]^\top$$

is called the difference between $\boldsymbol{a}$ and $\boldsymbol{b}$ and is denoted $\boldsymbol{a} - \boldsymbol{b}$.

The vector $\boldsymbol{0} - \boldsymbol{b}$ is denoted $-\boldsymbol{b}$. Note that

$$\begin{aligned} \boldsymbol{b} + (\boldsymbol{a} - \boldsymbol{b}) &= \boldsymbol{a}, \\ -(-\boldsymbol{b}) &= \boldsymbol{b}, \\ -(\boldsymbol{a} - \boldsymbol{b}) &= \boldsymbol{b} - \boldsymbol{a}. \end{aligned}$$

The vector $\boldsymbol{b} - \boldsymbol{a}$ is the unique solution of the vector equation

$$\boldsymbol{a} + \boldsymbol{x} = \boldsymbol{b}.$$

Indeed, suppose that $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^\top$ is a solution to $\boldsymbol{a} + \boldsymbol{x} = \boldsymbol{b}$. Then,

$$\begin{aligned} a_1 + x_1 &= b_1, \\ a_2 + x_2 &= b_2, \\ &\vdots \\ a_n + x_n &= b_n, \end{aligned}$$

and thus

$$\boldsymbol{x} = \boldsymbol{b} - \boldsymbol{a}.$$

We define an operation of multiplication of a vector $\boldsymbol{a} \in \mathbb{R}^n$ by a real scalar $\alpha \in \mathbb{R}$ as

$$\alpha\boldsymbol{a} = [\alpha a_1, \alpha a_2, \ldots, \alpha a_n]^\top.$$

This operation has the following properties:

1. The operation is distributive: for any real scalars α and β,

$$\begin{aligned} \alpha(\boldsymbol{a} + \boldsymbol{b}) &= \alpha\boldsymbol{a} + \alpha\boldsymbol{b}, \\ (\alpha + \beta)\boldsymbol{a} &= \alpha\boldsymbol{a} + \beta\boldsymbol{a}. \end{aligned}$$

2. The operation is associative:

$$\alpha(\beta\boldsymbol{a}) = (\alpha\beta)\boldsymbol{a}.$$

3. The scalar 1 satisfies

$$1\boldsymbol{a} = \boldsymbol{a}.$$

4. Any scalar α satisfies

$$\alpha\boldsymbol{0} = \boldsymbol{0}.$$

5. The scalar 0 satisfies

$$0\boldsymbol{a} = \boldsymbol{0}.$$

6. The scalar -1 satisfies

$$(-1)\boldsymbol{a} = -\boldsymbol{a}.$$

Note that $\alpha\boldsymbol{a} = \boldsymbol{0}$ if and only if $\alpha = 0$ or $\boldsymbol{a} = \boldsymbol{0}$. To see this, observe that $\alpha\boldsymbol{a} = \boldsymbol{0}$ is equivalent to $\alpha a_1 = \alpha a_2 = \cdots = \alpha a_n = 0$. If $\alpha = 0$ or $\boldsymbol{a} = \boldsymbol{0}$, then $\alpha\boldsymbol{a} = \boldsymbol{0}$. If $\boldsymbol{a} \neq \boldsymbol{0}$, then at least one of its components $a_k \neq 0$. For this component, $\alpha a_k = 0$, and hence we must have $\alpha = 0$. Similar arguments can be applied to the case when $\alpha \neq 0$.

A set of vectors $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k\}$ is said to be *linearly independent* if the equality

$$\alpha_1 \boldsymbol{a}_1 + \alpha_2 \boldsymbol{a}_2 + \cdots + \alpha_k \boldsymbol{a}_k = \mathbf{0}$$

implies that all coefficients α_i, $i = 1, \ldots, k$, are equal to zero. A set of the vectors $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k\}$ is *linearly dependent* if it is not linearly independent.

Note that the set composed of the single vector $\mathbf{0}$ is linearly dependent, for if $\alpha \neq 0$, then $\alpha \mathbf{0} = \mathbf{0}$. In fact, any set of vectors containing the vector $\mathbf{0}$ is linearly dependent.

A set composed of a single nonzero vector $\boldsymbol{a} \neq \mathbf{0}$ is linearly independent since $\alpha \boldsymbol{a} = \mathbf{0}$ implies that $\alpha = 0$.

A vector $\boldsymbol{a}$ is said to be a *linear combination* of vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k$ if there are scalars $\alpha_1, \ldots, \alpha_k$ such that

$$\boldsymbol{a} = \alpha_1 \boldsymbol{a}_1 + \alpha_2 \boldsymbol{a}_2 + \cdots + \alpha_k \boldsymbol{a}_k.$$

Proposition 2.1 *A set of vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k\}$ is linearly dependent if and only if one of the vectors from the set is a linear combination of the remaining vectors.* □

Proof. $\Rightarrow$: If $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k\}$ is linearly dependent, then

$$\alpha_1 \boldsymbol{a}_1 + \alpha_2 \boldsymbol{a}_2 + \cdots + \alpha_k \boldsymbol{a}_k = \mathbf{0},$$

where at least one of the scalars $\alpha_i \neq 0$, whence

$$\boldsymbol{a}_i = -\frac{\alpha_1}{\alpha_i} \boldsymbol{a}_1 - \frac{\alpha_2}{\alpha_i} \boldsymbol{a}_2 - \cdots - \frac{\alpha_k}{\alpha_i} \boldsymbol{a}_k.$$

$\Leftarrow$: Suppose that

$$\boldsymbol{a}_1 = \alpha_2 \boldsymbol{a}_2 + \alpha_3 \boldsymbol{a}_3 + \cdots + \alpha_k \boldsymbol{a}_k,$$

then

$$(-1)\boldsymbol{a}_1 + \alpha_2 \boldsymbol{a}_2 + \cdots + \alpha_k \boldsymbol{a}_k = \mathbf{0}.$$

Because the first scalar is nonzero, the set of vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k\}$ is linearly dependent. The same argument holds if $\boldsymbol{a}_i$, $i = 2, \ldots, k$, is a linear combination of the remaining vectors. ∎

A subset $\mathcal{V}$ of $\mathbb{R}^n$ is called a *subspace* of $\mathbb{R}^n$ if $\mathcal{V}$ is closed under the operations of vector addition and scalar multiplication. That is, if $\boldsymbol{a}$ and $\boldsymbol{b}$ are vectors in $\mathcal{V}$, then the vectors $\boldsymbol{a} + \boldsymbol{b}$ and $\alpha \boldsymbol{a}$ are also in $\mathcal{V}$ for every scalar α.

Every subspace contains the zero vector $\mathbf{0}$, for if $\boldsymbol{a}$ is an element of the subspace, so is $(-1)\boldsymbol{a} = -\boldsymbol{a}$. Hence, $\boldsymbol{a} - \boldsymbol{a} = \mathbf{0}$ also belongs to the subspace.

Let $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k$ be arbitrary vectors in $\mathbb{R}^n$. The set of all their linear combinations is called the *span* of $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k$ and is denoted

$$\operatorname{span}[\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k] = \left\{ \sum_{i=1}^{k} \alpha_i \boldsymbol{a}_i : \alpha_1, \ldots, \alpha_k \in \mathbb{R} \right\}.$$

Given a vector $\boldsymbol{a}$, the subspace span$[\boldsymbol{a}]$ is composed of the vectors $\alpha\boldsymbol{a}$, where α is an arbitrary real number ($\alpha \in \mathbb{R}$). Also observe that if $\boldsymbol{a}$ is a linear combination of $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k$, then

$$\operatorname{span}[\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k, \boldsymbol{a}] = \operatorname{span}[\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k].$$

The span of any set of vectors is a subspace.

Given a subspace $\mathcal{V}$, any set of linearly independent vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k\} \subset \mathcal{V}$ such that $\mathcal{V} = \operatorname{span}[\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k]$ is referred to as a *basis* of the subspace $\mathcal{V}$. All bases of a subspace $\mathcal{V}$ contain the same number of vectors. This number is called the *dimension* of $\mathcal{V}$, denoted $\dim \mathcal{V}$.

Proposition 2.2 *If $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k\}$ is a basis of $\mathcal{V}$, then any vector $\boldsymbol{a}$ of $\mathcal{V}$ can be represented uniquely as*

$$\boldsymbol{a} = \alpha_1\boldsymbol{a}_1 + \alpha_2\boldsymbol{a}_2 + \cdots + \alpha_k\boldsymbol{a}_k,$$

where $\alpha_i \in \mathbb{R}$, $i = 1, 2, \ldots, k$. □

Proof. To prove the uniqueness of the representation of $\boldsymbol{a}$ in terms of the basis vectors, assume that

$$\boldsymbol{a} = \alpha_1\boldsymbol{a}_1 + \alpha_2\boldsymbol{a}_2 + \cdots + \alpha_k\boldsymbol{a}_k$$

and

$$\boldsymbol{a} = \beta_1\boldsymbol{a}_1 + \beta_2\boldsymbol{a}_2 + \cdots + \beta_k\boldsymbol{a}_k.$$

We now show that $\alpha_i = \beta_i$, $i = 1, \ldots, k$. We have

$$\alpha_1\boldsymbol{a}_1 + \alpha_2\boldsymbol{a}_2 + \cdots + \alpha_k\boldsymbol{a}_k = \beta_1\boldsymbol{a}_1 + \beta_2\boldsymbol{a}_2 + \cdots + \beta_k\boldsymbol{a}_k$$

or

$$(\alpha_1 - \beta_1)\boldsymbol{a}_1 + (\alpha_2 - \beta_2)\boldsymbol{a}_2 + \cdots + (\alpha_k - \beta_k)\boldsymbol{a}_k = \boldsymbol{0}.$$

Because the set $\{\boldsymbol{a}_i : i = 1, 2, \ldots, k\}$ is linearly independent, $\alpha_1 - \beta_1 = \alpha_2 - \beta_2 = \cdots = \alpha_k - \beta_k = 0$, which implies that $\alpha_i = \beta_i$, $i = 1, \ldots, k$. ■

Suppose that we are given a basis $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k\}$ of $\mathcal{V}$ and a vector $\boldsymbol{a} \in \mathcal{V}$ such that

$$\boldsymbol{a} = \alpha_1\boldsymbol{a}_1 + \alpha_2\boldsymbol{a}_2 + \cdots + \alpha_k\boldsymbol{a}_k.$$

The coefficients α_i, $i = 1, \ldots, k$, are called the *coordinates* of $\boldsymbol{a}$ with respect to the basis $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k\}$.

The *natural basis* for $\mathbb{R}^n$ is the set of vectors

$$\boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \ldots, \quad \boldsymbol{e}_n = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

The reason for calling these vectors the natural basis is that

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 + \cdots + x_n\boldsymbol{e}_n.$$

We can similarly define *complex vector spaces.* For this, let $\mathbb{C}$ denote the set of complex numbers and $\mathbb{C}^n$ the set of column n-vectors with complex components. As the reader can easily verify, the set $\mathbb{C}^n$ has properties similar to those of $\mathbb{R}^n$, where scalars can take complex values.

A *matrix* is a rectangular array of numbers, commonly denoted by uppercase bold letters (e.g., $\boldsymbol{A}$). A matrix with m rows and n columns is called an $m \times n$ matrix, and we write

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

The real number a_{ij} located in the ith row and jth column is called the (i,j)th *entry.* We can think of $\boldsymbol{A}$ in terms of its n columns, each of which is a column vector in $\mathbb{R}^m$. Alternatively, we can think of $\boldsymbol{A}$ in terms of its m rows, each of which is a row n-vector.

Consider the $m \times n$ matrix $\boldsymbol{A}$ above. The *transpose* of matrix $\boldsymbol{A}$, denoted $\boldsymbol{A}^\top$, is the $n \times m$ matrix

$$\boldsymbol{A}^\top = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix};$$

that is, the columns of $\boldsymbol{A}$ are the rows of $\boldsymbol{A}^\top$, and vice versa.

Let the symbol $\mathbb{R}^{m\times n}$ denote the set of $m\times n$ matrices whose entries are real numbers. We treat column vectors in $\mathbb{R}^n$ as elements of $\mathbb{R}^{n\times 1}$. Similarly, we treat row n-vectors as elements of $\mathbb{R}^{1\times n}$. Accordingly, vector transposition is simply a special case of matrix transposition, and we will no longer distinguish between the two. Note that there is a slight inconsistency in the notation of row vectors when identified as $1 \times n$ matrices: We separate the components of the row vector with commas, whereas in matrix notation we do not generally use commas. However, the use of commas in separating elements in a row helps to clarify their separation. We use use such commas even in separating matrices arranged in a horizontal row.