Cambridge University Press & Assessment
978-1-316-51896-0 — Introduction to Applied Linear Algebra
Stephen Boyd, Lieven Vandenberghe
Excerpt
More Information

Part I

Vectors

Chapter 1

Vectors

In this chapter we introduce vectors and some common operations on them. We describe some settings in which vectors are used.

1.1 Vectors

A *vector* is an ordered finite list of numbers. Vectors are usually written as vertical arrays, surrounded by square or curved brackets, as in

$$\begin{bmatrix} -1.1 \\ 0.0 \\ 3.6 \\ -7.2 \end{bmatrix} \quad \text{or} \quad \begin{pmatrix} -1.1 \\ 0.0 \\ 3.6 \\ -7.2 \end{pmatrix}.$$

They can also be written as numbers separated by commas and surrounded by parentheses. In this notation style, the vector above is written as

$$(-1.1, 0.0, 3.6, -7.2).$$

The *elements* (or *entries*, *coefficients*, *components*) of a vector are the values in the array. The *size* (also called *dimension* or *length*) of the vector is the number of elements it contains. The vector above, for example, has size four; its third entry is 3.6. A vector of size n is called an n*-vector*. A 1-vector is considered to be the same as a number, *i.e.*, we do not distinguish between the 1-vector $[\,1.3\,]$ and the number 1.3.

We often use symbols to denote vectors. If we denote an n-vector using the symbol a, the ith element of the vector a is denoted a_i, where the subscript i is an integer index that runs from 1 to n, the size of the vector.

Two vectors a and b are *equal*, which we denote $a = b$, if they have the same size, and each of the corresponding entries is the same. If a and b are n-vectors, then $a = b$ means $a_1 = b_1, \ldots, a_n = b_n$.

The numbers or values of the elements in a vector are called *scalars*. We will focus on the case that arises in most applications, where the scalars are real numbers. In this case we refer to vectors as *real vectors*. (Occasionally other types of scalars arise, for example, complex numbers, in which case we refer to the vector as a *complex vector*.) The set of all real numbers is written as $\mathbf{R}$, and the set of all real n-vectors is denoted $\mathbf{R}^n$, so $a \in \mathbf{R}^n$ is another way to say that a is an n-vector with real entries. Here we use set notation: $a \in \mathbf{R}^n$ means that a is an element of the set $\mathbf{R}^n$; see appendix A.

Block or stacked vectors. It is sometimes useful to define vectors by *concatenating* or *stacking* two or more vectors, as in

$$a = \begin{bmatrix} b \\ c \\ d \end{bmatrix},$$

where a, b, c, and d are vectors. If b is an m-vector, c is an n-vector, and d is a p-vector, this defines the $(m+n+p)$-vector

$$a = (b_1, b_2, \ldots, b_m, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_p).$$

The stacked vector a is also written as $a = (b, c, d)$.

Stacked vectors can include scalars (numbers). For example if a is a 3-vector, $(1, a)$ is the 4-vector $(1, a_1, a_2, a_3)$.

Subvectors. In the equation above, we say that b, c, and d are *subvectors* or *slices* of a, with sizes m, n, and p, respectively. *Colon notation* is used to denote subvectors. If a is a vector, then $a_{r:s}$ is the vector of size $s - r + 1$, with entries $a_r, \ldots, a_s$:

$$a_{r:s} = (a_r, \ldots, a_s).$$

The subscript $r\!:\!s$ is called the *index range*. Thus, in our example above, we have

$$b = a_{1:m}, \qquad c = a_{(m+1):(m+n)}, \qquad d = a_{(m+n+1):(m+n+p)}.$$

As a more concrete example, if z is the 4-vector $(1, -1, 2, 0)$, the slice $z_{2:3}$ is $z_{2:3} = (-1, 2)$. Colon notation is not completely standard, but it is growing in popularity.

Notational conventions. Some authors try to use notation that helps the reader distinguish between vectors and scalars (numbers). For example, Greek letters (α, β, ...) might be used for numbers, and lower-case letters (a, x, f, ...) for vectors. Other notational conventions include vectors given in bold font ($\mathbf{g}$), or vectors written with arrows above them ($\vec{a}$). These notational conventions are not standardized, so you should be prepared to figure out what things are (*i.e.*, scalars or vectors) despite the author's notational scheme (if any exists).

Indexing. We should give a couple of warnings concerning the subscripted index notation a_i. The first warning concerns the range of the index. In many computer languages, arrays of length n are indexed from $i = 0$ to $i = n - 1$. But in standard mathematical notation, n-vectors are indexed from $i = 1$ to $i = n$, so in this book, vectors will be indexed from $i = 1$ to $i = n$.

The next warning concerns an ambiguity in the notation a_i, used for the ith element of a vector a. The same notation will occasionally refer to the ith vector in a collection or list of k vectors $a_1, \ldots, a_k$. Whether a_3 means the third element of a vector a (in which case a_3 is a number), or the third vector in some list of vectors (in which case a_3 is a vector) should be clear from the context. When we need to refer to an element of a vector that is in an indexed collection of vectors, we can write $(a_i)_j$ to refer to the jth entry of a_i, the ith vector in our list.

Zero vectors. A *zero vector* is a vector with all elements equal to zero. Sometimes the zero vector of size n is written as 0_n, where the subscript denotes the size. But usually a zero vector is denoted just 0, the same symbol used to denote the number 0. In this case you have to figure out the size of the zero vector from the context. As a simple example, if a is a 9-vector, and we are told that $a = 0$, the 0 vector on the right-hand side must be the one of size 9.

Even though zero vectors of different sizes are different vectors, we use the same symbol 0 to denote them. In computer programming this is called *overloading*: The symbol 0 is overloaded because it can mean different things depending on the context (*e.g.*, the equation it appears in).

Unit vectors. A (standard) *unit vector* is a vector with all elements equal to zero, except one element which is equal to one. The ith unit vector (of size n) is the unit vector with ith element one, and denoted e_i. For example, the vectors

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

are the three unit vectors of size 3. The notation for unit vectors is an example of the ambiguity in notation noted above. Here, e_i denotes the ith unit vector, and not the ith element of a vector e. Thus we can describe the ith unit n-vector e_i as

$$(e_i)_j = \begin{cases} 1 & j = i \\ 0 & j \neq i, \end{cases}$$

for $i, j = 1, \ldots, n$. On the left-hand side e_i is an n-vector; $(e_i)_j$ is a number, its jth entry. As with zero vectors, the size of e_i is usually determined from the context.

Ones vector. We use the notation $\mathbf{1}_n$ for the n-vector with all its elements equal to one. We also write $\mathbf{1}$ if the size of the vector can be determined from the context. (Some authors use e to denote a vector of all ones, but we will not use this notation.) The vector $\mathbf{1}$ is sometimes called the *ones vector*.

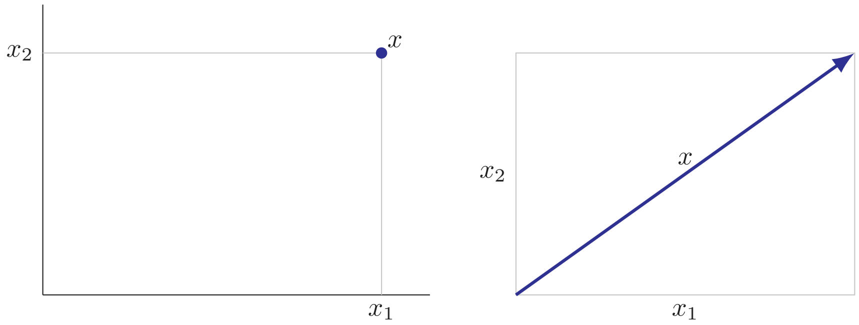

Figure 1.1 *Left.* The 2-vector x specifies the position (shown as a dot) with coordinates x_1 and x_2 in a plane. *Right.* The 2-vector x represents a displacement in the plane (shown as an arrow) by x_1 in the first axis and x_2 in the second.

Sparsity. A vector is said to be *sparse* if many of its entries are zero; its *sparsity pattern* is the set of indices of nonzero entries. The number of the nonzero entries of an n-vector x is denoted $\mathbf{nnz}(x)$. Unit vectors are sparse, since they have only one nonzero entry. The zero vector is the sparsest possible vector, since it has no nonzero entries. Sparse vectors arise in many applications.

Examples

An n-vector can be used to represent n quantities or values in an application. In some cases the values are similar in nature (for example, they are given in the same physical units); in others, the quantities represented by the entries of the vector are quite different from each other. We briefly describe below some typical examples, many of which we will see throughout the book.

Location and displacement. A 2-vector can be used to represent a position or location in a 2-dimensional (2-D) space, *i.e.*, a plane, as shown in figure 1.1. A 3-vector is used to represent a location or position of some point in 3-dimensional (3-D) space. The entries of the vector give the coordinates of the position or location.

A vector can also be used to represent a displacement in a plane or 3-D space, in which case it is typically drawn as an arrow, as shown in figure 1.1. A vector can also be used to represent the velocity or acceleration, at a given time, of a point that moves in a plane or 3-D space.

Color. A 3-vector can represent a color, with its entries giving the Red, Green, and Blue (RGB) intensity values (often between 0 and 1). The vector $(0, 0, 0)$ represents black, the vector $(0, 1, 0)$ represents a bright pure green color, and the vector $(1, 0.5, 0.5)$ represents a shade of pink. This is illustrated in figure 1.2.

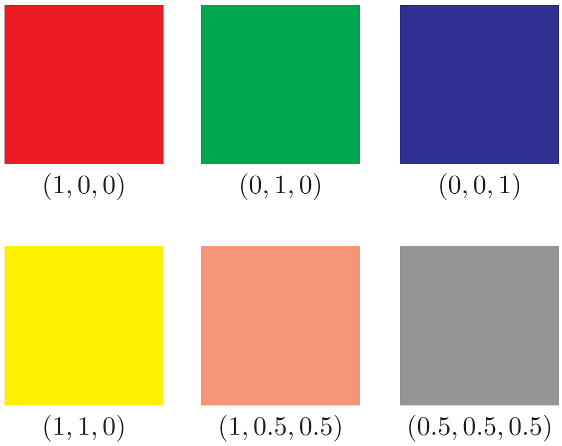

Figure 1.2 Six colors and their RGB vectors.

Quantities. An n-vector q can represent the amounts or quantities of n different resources or products held (or produced, or required) by an entity such as a company. Negative entries mean an amount of the resource owed to another party (or consumed, or to be disposed of). For example, a *bill of materials* is a vector that gives the amounts of n resources required to create a product or carry out a task.

Portfolio. An n-vector s can represent a stock portfolio or investment in n different assets, with s_i giving the number of shares of asset i held. The vector $(100, 50, 20)$ represents a portfolio consisting of 100 shares of asset 1, 50 shares of asset 2, and 20 shares of asset 3. Short positions (*i.e.*, shares that you owe another party) are represented by negative entries in a portfolio vector. The entries of the portfolio vector can also be given in dollar values, or fractions of the total dollar amount invested.

Values across a population. An n-vector can give the values of some quantity across a population of individuals or entities. For example, an n-vector b can give the blood pressure of a collection of n patients, with b_i the blood pressure of patient i, for $i = 1, \ldots, n$.

Proportions. A vector w can be used to give fractions or proportions out of n choices, outcomes, or options, with w_i the fraction with choice or outcome i. In this case the entries are nonnegative and add up to one. Such vectors can also be interpreted as the recipes for a mixture of n items, an allocation across n entities, or as probability values in a probability space with n outcomes. For example, a uniform mixture of 4 outcomes is represented as the 4-vector $(1/4, 1/4, 1/4, 1/4)$.

Time series. An n-vector can represent a *time series* or *signal*, that is, the value of some quantity at different times. (The entries in a vector that represents a time series are sometimes called *samples*, especially when the quantity is something

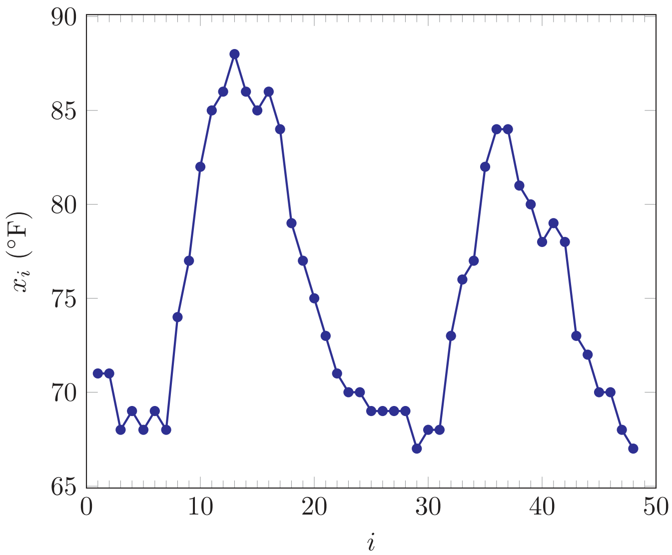

Figure 1.3 Hourly temperature in downtown Los Angeles on August 5 and 6, 2015 (starting at 12:47AM, ending at 11:47PM).

measured.) An audio (sound) signal can be represented as a vector whose entries give the value of acoustic pressure at equally spaced times (typically 48000 or 44100 per second). A vector might give the hourly rainfall (or temperature, or barometric pressure) at some location, over some time period. When a vector represents a time series, it is natural to plot x_i versus i with lines connecting consecutive time series values. (These lines carry no information; they are added only to make the plot easier to understand visually.) An example is shown in figure 1.3, where the 48-vector x gives the hourly temperature in downtown Los Angeles over two days.

Daily return. A vector can represent the daily return of a stock, *i.e.*, its fractional increase (or decrease if negative) in value over the day. For example the return time series vector $(-0.022, +0.014, +0.004)$ means the stock price went down 2.2% on the first day, then up 1.4% the next day, and up again 0.4% on the third day. In this example, the samples are not uniformly spaced in time; the index refers to trading days, and does not include weekends or market holidays. A vector can represent the daily (or quarterly, hourly, or minute-by-minute) value of any other quantity of interest for an asset, such as price or volume.

Cash flow. A cash flow into and out of an entity (say, a company) can be represented by a vector, with positive entries representing payments to the entity, and negative entries representing payments by the entity. For example, with entries giving cash flow each quarter, the vector $(1000, -10, -10, -10, -1010)$ represents a one year loan of \$1000, with 1% interest only payments made each quarter, and the principal and last interest payment at the end.

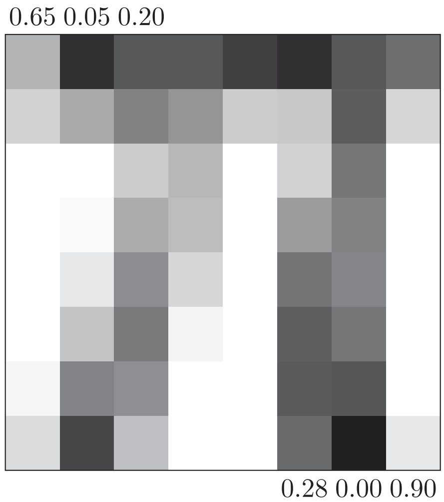

Figure 1.4 8×8 image and the grayscale levels at six pixels.

Images. A monochrome (black and white) image is an array of $M \times N$ pixels (square patches with uniform grayscale level) with M rows and N columns. Each of the MN pixels has a grayscale or intensity value, with 0 corresponding to black and 1 corresponding to bright white. (Other ranges are also used.) An image can be represented by a vector of length MN, with the elements giving grayscale levels at the pixel locations, typically ordered column-wise or row-wise.

Figure 1.4 shows a simple example, an 8×8 image. (This is a very low resolution; typical values of M and N are in the hundreds or thousands.) With the vector entries arranged row-wise, the associated 64-vector is

$$x = (0.65, 0.05, 0.20, \ldots, 0.28, 0.00, 0.90).$$

A color $M \times N$ pixel image is described by a vector of length $3MN$, with the entries giving the R, G, and B values for each pixel, in some agreed-upon order.

Video. A monochrome video, *i.e.*, a sequence of length K of images with $M \times N$ pixels, can be represented by a vector of length KMN (again, in some particular order).

Word count and histogram. A vector of length n can represent the number of times each word in a dictionary of n words appears in a document. For example, $(25, 2, 0)$ means that the first dictionary word appears 25 times, the second one twice, and the third one not at all. (Typical dictionaries used for document word counts have many more than 3 elements.) A small example is shown in figure 1.5. A variation is to have the entries of the vector give the *histogram* of word frequencies in the document, so that, *e.g.*, $x_5 = 0.003$ means that 0.3% of all the words in the document are the fifth word in the dictionary.

It is common practice to count variations of a word (say, the same word stem with different endings) as the same word; for example, 'rain', 'rains', 'raining', and

Word count vectors are used in computer based document analysis. Each entry of the word count vector is the number of times the associated dictionary word appears in the document.

word	3
in	2
number	1
horse	0
the	4
document	2

Figure 1.5 A snippet of text (top), the dictionary (bottom left), and word count vector (bottom right).

'rained' might all be counted as 'rain'. Reducing each word to its stem is called *stemming*. It is also common practice to exclude words that are too common (such as 'a' or 'the'), which are referred to as *stop words*, as well as words that are extremely rare.

Customer purchases. An n-vector p can be used to record a particular customer's purchases from some business over some period of time, with p_i the quantity of item i the customer has purchased, for $i = 1, \ldots, n$. (Unless n is small, we would expect many of these entries to be zero, meaning the customer has not purchased those items.) In one variation, p_i represents the total dollar value of item i the customer has purchased.

Occurrence or subsets. An n-vector o can be used to record whether or not each of n different events has occurred, with $o_i = 0$ meaning that event i did not occur, and $o_i = 1$ meaning that it did occur. Such a vector encodes a subset of a collection of n objects, with $o_i = 1$ meaning that object i is contained in the subset, and $o_i = 0$ meaning that object i is not in the subset. Each entry of the vector a is either 0 or 1; such vectors are called *Boolean*, after the mathematician George Boole, a pioneer in the study of logic.

Features or attributes. In many applications a vector collects together n different quantities that pertain to a single thing or object. The quantities can be measurements, or quantities that can be measured or derived from the object. Such a vector is sometimes called a *feature vector*, and its entries are called the *features* or *attributes*. For example, a 6-vector f could give the age, height, weight, blood pressure, temperature, and gender of a patient admitted to a hospital. (The last entry of the vector could be encoded as $f_6 = 0$ for male, $f_6 = 1$ for female.) In this example, the quantities represented by the entries of the vector are quite different, with different physical units.

Vector entry labels. In applications such as the ones described above, each entry of a vector has a meaning, such as the count of a specific word in a document, the number of shares of a specific stock held in a portfolio, or the rainfall in a specific hour. It is common to keep a separate list of labels or tags that explain or annotate the meaning of the vector entries. As an example, we might associate the portfolio vector $(100, 50, 20)$ with the list of ticker symbols (AAPL, INTC, AMZN), so we know that assets 1, 2, and 3 are Apple, Intel, and Amazon. In some applications, such as an image, the meaning or ordering of the entries follow known conventions or standards.

1.2 Vector addition

Two vectors *of the same size* can be added together by adding the corresponding elements, to form another vector of the same size, called the *sum* of the vectors. Vector addition is denoted by the symbol $+$. (Thus the symbol $+$ is overloaded to mean scalar addition when scalars appear on its left- and right-hand sides, and vector addition when vectors appear on its left- and right-hand sides.) For example,

$$\begin{bmatrix} 0 \\ 7 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 9 \\ 3 \end{bmatrix}.$$

Vector subtraction is similar. As an example,

$$\begin{bmatrix} 1 \\ 9 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \end{bmatrix}.$$

The result of vector subtraction is called the *difference* of the two vectors.

Properties. Several properties of vector addition are easily verified. For any vectors a, b, and c of the same size we have the following.

- Vector addition is *commutative*: $a + b = b + a$.
- Vector addition is *associative*: $(a + b) + c = a + (b + c)$. We can therefore write both as $a + b + c$.
- $a + 0 = 0 + a = a$. Adding the zero vector to a vector has no effect. (This is an example where the size of the zero vector follows from the context: It must be the same as the size of a.)
- $a - a = 0$. Subtracting a vector from itself yields the zero vector. (Here too the size of 0 is the size of a.)

To show that these properties hold, we argue using the definition of vector addition and vector equality. As an example, let us show that for any n-vectors a and b, we have $a + b = b + a$. The ith entry of $a + b$ is, by the definition of vector