APPLIED LINEAR ALGEBRA AND NUMERICAL ANALYSIS

AMATH 352, Winter 2017

Room: MWF 12:30-1:20 GWN 301

Instructor: Brian de Silva

TA: Matthew Farrell

This course covers basic concepts of linear algebra, with an emphasis on computational techniques. Linear algebra plays a fundamental role in a wide range of applications from physical and social sciences, statistics, engineering, finance, computer graphics, big data and machine learning.

In this course, we will study vectors, vector spaces, linear transformations, matrix-vector manipulations, solving linear systems, least squares problems, and eigenvalue problems. Matrix decompositions (e.g. LU, QR, SVD, etc.) play a fundamental role in the course.

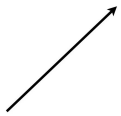
These notes were written by Professor Randall J. LeVeque, with additional contributions by Professor Ulrich Hetmaniuk. I will continue to modify them as we proceed through the course. The latest version will be available from the course website.

Contents

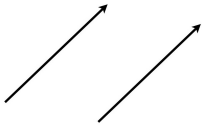
List of Figures

1 Column Vectors

In geometry, the two-dimensional plane is denoted $\mathbb{R}^2$ and the three-dimensional space $\mathbb{R}^3$. A vector is an object comprised of a magnitude and a direction. In the plane, we can draw a vector as an arrow with some length and pointing somewhere . A vector can also be thought of as a displacement. A displacement



does not depend where it starts. Consequently, the vectors are equal, even



though they start from different places, because they have equal length and equal direction. The basic idea here, combining magnitude with direction, is the key to extending to higher dimensions. In this section, we define the generalization of vectors in the two-dimensional plane and three-dimensional space.

Let m be a positive integer. We denote $\mathbb{R}^m$ the set of all real m-tuples, i.e. the set of all sequences with m components, each of which is a real number. The standard notation for an element $\mathbf{x}$ of $\mathbb{R}^m$ is the column vector notation:

$$\forall \mathbf{x} \in \mathbb{R}^m \Longleftrightarrow \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}. \tag{1.1}$$

It is important to remember that, in many applications of linear algebra, the elements of the vector represent something different from the three physical coordinates of ordinary space. There is often nothing unphysical about considering vectors with many more than 3 components.

Example 1. We have

$$\begin{bmatrix} -1 \\ 3 \end{bmatrix} \in \mathbb{R}^2, \quad \begin{bmatrix} \sqrt{7} \\ 0 \\ \sqrt{3} \end{bmatrix} \in \mathbb{R}^3, \text{ and } \begin{bmatrix} 1 \\ -2/5 \\ -3/5 \\ 4 \end{bmatrix} \in \mathbb{R}^4.$$

Example 2. The vector with entries equal to 0 is denoted $\mathbf{0}$. For example, we

have

$$\mathbf{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^2 \text{ and } \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^4.$$

1.1 Addition of column vectors

We can define the addition of two vectors $\mathbf{x}$ and $\mathbf{y}$ of $\mathbb{R}^m$:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}, \quad \mathbf{x}+\mathbf{y} = \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_m + y_m \end{bmatrix}. \tag{1.2}$$

The set $\mathbb{R}^m$ is closed under addition, meaning that whenever the addition is applied to vectors in $\mathbb{R}^m$, we obtain another vector in the same set $\mathbb{R}^m$. For example, we have

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 \\ -4 \\ 8 \end{bmatrix} = \begin{bmatrix} 1+2 \\ 2-4 \\ 3+8 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \\ 11 \end{bmatrix}.$$

1.2 Scalar multiplication of a column vector

We can also define the scalar multiplication: if $\mathbf{x} \in \mathbb{R}^m$ and $\alpha \in \mathbb{R}$, then the vector $\alpha\mathbf{x}$ belongs to $\mathbb{R}^m$ and is defined by multiplying each component of $\mathbf{x}$ by the scalar α:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \quad \alpha \in \mathbb{R}, \quad \alpha\mathbf{x} = \begin{bmatrix} \alpha x_1 \\ \vdots \\ \alpha x_m \end{bmatrix}. \tag{1.3}$$

The set $\mathbb{R}^m$ is also closed under scalar multiplication. For example, we have

$$2\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \times 1 \\ 2 \times 2 \\ 2 \times 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}, \quad (-3)\begin{bmatrix} 2 \\ -4 \\ 8 \end{bmatrix} = \begin{bmatrix} (-3) \times 2 \\ (-3) \times (-4) \\ (-3) \times 8 \end{bmatrix} = \begin{bmatrix} -6 \\ 12 \\ -24 \end{bmatrix}.$$

1.3 Norms

To measure the magnitude or the length of a vector, we use a vector norm. A vector norm is simply a function (or map) taking vectors in $\mathbb{R}^m$ as input and returning nonnegative real numbers as output. By definition, a vector norm satisfies the following conditions (which generalize important properties of the

absolute value for scalars):

$$\forall \mathbf{x} \in \mathbb{R}^m, \|\mathbf{x}\| \geq 0, \tag{1.4a}$$
$$\text{if } \|\mathbf{x}\| = 0, \text{ then } \mathbf{x} = \mathbf{0}, \tag{1.4b}$$
$$\text{if } \mathbf{x} = \mathbf{0}, \text{ then } \|\mathbf{x}\| = 0, \tag{1.4c}$$
$$\forall \alpha \in \mathbb{R}, \|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|, \tag{1.4d}$$
$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^m, \|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \quad \text{(Triangle inequality)}. \tag{1.4e}$$

Note that a norm satisfies the following properties

$$\|-\mathbf{x}\| = \|\mathbf{x}\| \quad \text{and} \quad \|\mathbf{x}+\mathbf{x}\| = \|2\mathbf{x}\| = 2\|\mathbf{x}\|.$$

The triangle inequality can not be an equality because we have, for any non-zero vector $\mathbf{x}$,

$$0 = \|\mathbf{x}-\mathbf{x}\| < \|\mathbf{x}\| + \|-\mathbf{x}\| = 2\|\mathbf{x}\|.$$

One common choice is the max-norm (or infinity-norm) denoted by

$$\|\mathbf{x}\|_\infty = \max_{1\leq i\leq m} |x_i|. \tag{1.5}$$

A bound on the max-norm of the error is nice because we know that every component of the error can be no greater than the max-norm,

$$\forall\, i,\ 1 \leq i \leq m,\ |x_i| \leq \|\mathbf{x}\|_\infty.$$

It is easy to verify that $\|\cdot\|_\infty$ satisfies the required properties (1.4).

- $\forall \mathbf{x} \in \mathbb{R}^m, \|\mathbf{x}\|_\infty \geq 0$ because the absolute value is always greater or equal to zero.
- $\|\mathbf{x}\|_\infty = 0$ if and only if $\mathbf{x} = \mathbf{0}$,
  - If $\mathbf{x} = \mathbf{0}$, then all the components of $\mathbf{x}$ are zero and, consequently, $\|\mathbf{x}\|_\infty = 0$.
  - If $\|\mathbf{x}\|_\infty = 0$, then every component x_i will be zero. This implies that the vector $\mathbf{x}$ is $\mathbf{0}$.
- $\forall \alpha \in \mathbb{R}, \|\alpha\mathbf{x}\|_\infty = |\alpha|\|\mathbf{x}\|_\infty$,
  - Note that, for every entry i, we have

    $$|\alpha x_i| = |\alpha||x_i|.$$

    We get

    $$\max_{1\leq i\leq m} |\alpha x_i| = \max_{1\leq i\leq m} |\alpha||x_i| = |\alpha| \max_{1\leq i\leq m} |x_i|$$

    since we are multiplying every entry by the same factor $|\alpha|$. We obtain

    $$\|\alpha\mathbf{x}\|_\infty = |\alpha|\|\mathbf{x}\|_\infty$$

- $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^m, \|\mathbf{x}+\mathbf{y}\|_\infty \leq \|\mathbf{x}\|_\infty + \|\mathbf{y}\|_\infty$
  - The absolute value satisfies a triangle inequality. Consequently, we have, for every i,

    $$|x_i + y_i| \leq |x_i| + |y_i| \leq \|\mathbf{x}\|_\infty + \|\mathbf{y}\|_\infty.$$

    So every entry in the vector $\mathbf{x}+\mathbf{y}$ is smaller than $\|\mathbf{x}\|_\infty + \|\mathbf{y}\|_\infty$. In particular, the largest entry will also satisfies that. So we get

    $$\max_{1\leq i\leq m} |x_i + y_i| = \|\mathbf{x}+\mathbf{y}\|_\infty \leq \|\mathbf{x}\|_\infty + \|\mathbf{y}\|_\infty$$

For some problems, however, there are other norms which are either more appropriate or easier to bound using our analytical tools.

The 2-norm is frequently used,

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{m} |x_i|^2}. \tag{1.6}$$

The 2-norm is often called the Euclidean norm. It is easy to verify that $\|\cdot\|_2$ satisfies the required properties (1.4).

- $\forall \mathbf{x} \in \mathbb{R}^m, \|\mathbf{x}\|_2 \geq 0$ because the sum of squares is non-negative and the square root is always greater or equal to zero.
- $\|\mathbf{x}\|_2 = 0$ if and only if $\mathbf{x} = \mathbf{0}$,
  - If $\mathbf{x} = \mathbf{0}$, then all the components of $\mathbf{x}$ are zero and, consequently, $\|\mathbf{x}\|_2 = 0$.
  - If $\|\mathbf{x}\|_2 = 0$, then every component x_i will be zero because we are summing non-negative numbers and the result is 0. This implies that the vector $\mathbf{x}$ is $\mathbf{0}$.
- $\forall \alpha \in \mathbb{R}, \|\alpha\mathbf{x}\|_2 = |\alpha|\|\mathbf{x}\|_2$,
  - Note that, for every entry i, we have

    $$|\alpha x_i|^2 = |\alpha|^2 |x_i|^2.$$

    We get

    $$\sum_{i=1}^{m} |\alpha x_i|^2 = \sum_{i=1}^{m} |\alpha|^2 |x_i|^2 = |\alpha|^2 \sum_{i=1}^{m} |x_i|^2.$$

    By taking the square root, we obtain

    $$\|\alpha\mathbf{x}\|_2 = |\alpha|\|\mathbf{x}\|_2.$$
- $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^m, \|\mathbf{x}+\mathbf{y}\|_2 \leq \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2$

– Because all the terms are positive, we can equivalently prove

$$\sum_{i=1}^{m}(x_i+y_i)^2 \leq \left(\sqrt{\sum_{i=1}^{m}x_i^2}+\sqrt{\sum_{i=1}^{m}y_i^2}\right)^2$$

or

$$\sum_{i=1}^{m}x_i^2+2\sum_{i=1}^{m}x_iy_i+\sum_{i=1}^{m}y_i^2 \leq \sum_{i=1}^{m}x_i^2+2\sqrt{\sum_{i=1}^{m}x_i^2}\sqrt{\sum_{i=1}^{m}y_i^2}+\sum_{i=1}^{m}y_i^2$$

or

$$2\sum_{i=1}^{m}x_iy_i \leq 2\sqrt{\sum_{i=1}^{m}x_i^2}\sqrt{\sum_{i=1}^{m}y_i^2}.$$

If $\mathbf{x}=\mathbf{0}$ or $\mathbf{y}=\mathbf{0}$, then the inequality is clear. So we can assume that $\mathbf{x}$ and $\mathbf{y}$ are non-zero. We would like to prove

$$\sum_{i=1}^{m}\frac{x_i}{\sqrt{\sum_{j=1}^{m}x_j^2}}\frac{y_i}{\sqrt{\sum_{j=1}^{m}y_j^2}} \leq 1.$$

To conclude, we recall the useful inequality

$$ab \leq \frac{1}{2}a^2+\frac{1}{2}b^2$$

which comes from the non-negativity of $(a-b)^2$.

$$\sum_{i=1}^{m}\frac{x_i}{\sqrt{\sum_{j=1}^{m}x_j^2}}\frac{y_i}{\sqrt{\sum_{j=1}^{m}y_j^2}} \leq \frac{1}{2}\sum_{i=1}^{m}\frac{x_i^2}{\sum_{j=1}^{m}x_j^2}+\frac{1}{2}\sum_{i=1}^{m}\frac{y_i^2}{\sum_{j=1}^{m}y_j^2}.$$

Note that

$$\sum_{i=1}^{m}\frac{x_i^2}{\sum_{j=1}^{m}x_j^2}=\frac{\sum_{i=1}^{m}x_i^2}{\sum_{j=1}^{m}x_j^2}=1 \quad \text{and} \quad \sum_{i=1}^{m}\frac{y_i^2}{\sum_{j=1}^{m}y_j^2}=\frac{\sum_{i=1}^{m}y_i^2}{\sum_{j=1}^{m}y_j^2}=1$$

which implies

$$\sum_{i=1}^{m}\frac{x_i}{\sqrt{\sum_{j=1}^{m}x_j^2}}\frac{y_i}{\sqrt{\sum_{j=1}^{m}y_j^2}} \leq \frac{1}{2}+\frac{1}{2} \leq 1.$$

This proves the triangle inequality.

The 1-norm is defined as follows

$$\|\mathbf{x}\|_1=\sum_{i=1}^{m}|x_i|. \tag{1.7}$$

The 1-norm is also known as the Manhattan norm because it corresponds to the distance traveled on a grid of city streets.

These norms are special cases of the general family of p-norms, defined by

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{m} |x_i|^p \right)^{\frac{1}{p}}. \tag{1.8}$$

Note that the max-norm can be obtained as the limit as $p \to +\infty$ of the p-norm.

For example, we have

$$\| \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \|_\infty = 3, \quad \| \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \|_1 = 6, \quad \| \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \|_2 = \sqrt{14},$$

and

$$\| \begin{bmatrix} -2 \\ 4 \\ -8 \end{bmatrix} \|_\infty = 8, \quad \| \begin{bmatrix} -2 \\ 4 \\ -8 \end{bmatrix} \|_1 = 14, \quad \| \begin{bmatrix} -2 \\ 4 \\ -8 \end{bmatrix} \|_2 = \sqrt{84}.$$

The closed unit ball $\{\mathbf{x} \in \mathbb{R}^m \mid \|\mathbf{x}\| \leq 1\}$ is the set of all vectors with a norm smaller than 1. The shape of this ball depends on the norm. The unit circle (or sphere) is the set of all vectors with a norm equal to 1,

$$S_1 = \{\mathbf{x} \in \mathbb{R}^m \mid \|\mathbf{x}\| = 1\}. \tag{1.9}$$

In $\mathbb{R}^2$, we can draw the unit circle as a curve composed of the points (x, y) such that

$$\| \begin{bmatrix} x \\ y \end{bmatrix} \| = 1.$$

There exists an infinite number of points on this curve. For example, for the Euclidian norm, the unit circle contains the vectors

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}, \begin{bmatrix} 1/2 \\ -\sqrt{3}/2 \end{bmatrix}, \dots$$

The equation governing the unit circle for the Euclidian norm is

$$\| \begin{bmatrix} x \\ y \end{bmatrix} \|_2 = 1 \quad \Rightarrow \quad \sqrt{x^2 + y^2} = 1.$$

Several ways are possible to draw this curve. Among them, we can parametrize x and y as follows

$$\begin{aligned} x &= \cos(\theta) \\ y &= \sin(\theta) \end{aligned}$$

where θ belongs to $[0, 2\pi]$. Figure 1.1 illustrates the unit circle for the Euclidian norm. The equation governing the unit circle for the 1-norm is

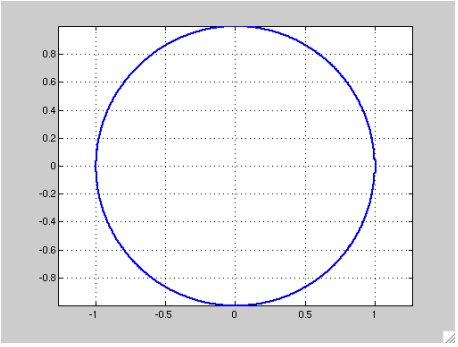


Figure 1.1: Unit circle for the Euclidian norm.

$$\| \begin{bmatrix} x \\ y \end{bmatrix} \|_1 = 1 \quad \Rightarrow \quad |x| + |y| = 1.$$

The curve is included in the square $[-1, 1] \times [-1, 1]$ and it is composed of the following four branches:

$$\begin{array}{rcll} x + y & = & 1 & 0 \leq x, y \leq 1 \\ x - y & = & 1 & x \in [0, 1] \text{ and } y \in [-1, 0] \\ -x - y & = & 1 & x, y \in [-1, 0] \\ -x + y & = & 1 & x \in [-1, 0] \text{ and } y \in [0, 1] \end{array}$$

Figure 1.2 illustrates the unit circle for the 1-norm.

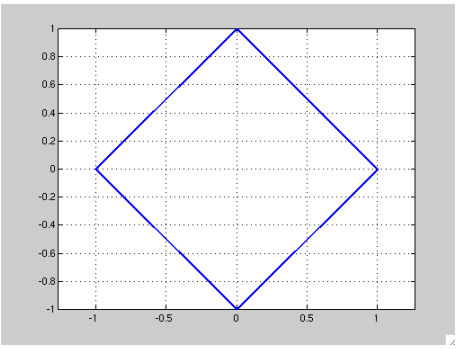


Figure 1.2: Unit circle for the 1-norm.

Trefethen and Bau[1] note that

> "The Sergel plaza in Stockholm, Sweden, has the shape of the unit ball in the 4-norm. The Danis poet Piet Hein popularized this *superellipse* as a pleasing shape for objects such as conference tables."

[1] L. N. Trefethen and D. Bau, *Numerical linear algebra*, SIAM, Philadelphia, 1997.

Other useful norms include the weighted p-norms, where each component of a vector is weighted. For example, a weighted 2-norm can be specified as follows

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{m} w_i |x_i|^2} \tag{1.10}$$

where the weights w_i are strictly positive real numbers.

For a given non-zero vector, its length depends on the norm chosen to measure it. However, in $\mathbb{R}^m$, all the norms are related. For example, we have

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{m}\|\mathbf{x}\|_\infty. \tag{1.11}$$

In $\mathbb{R}^3$, consider the vector

$$\mathbf{x} = \begin{bmatrix} 1 \\ -8 \\ 2 \end{bmatrix}.$$

Then we have

$$\begin{aligned} \| \begin{bmatrix} 1 \\ -8 \\ 2 \end{bmatrix} \|_\infty = 8 = \sqrt{8^2} \leq \sqrt{1^2 + (-8)^2 + 2^2} = \| \begin{bmatrix} 1 \\ -8 \\ 2 \end{bmatrix} \|_2 \\ \leq \sqrt{8^2 + 8^2 + 8^2} = \sqrt{3} \| \begin{bmatrix} 1 \\ -8 \\ 2 \end{bmatrix} \|_\infty \end{aligned} \tag{1.12}$$

where the first inequality is due to the fact that we add positive numbers to 8^2. The second inequality comes from the fact that any component is smaller than 8. To extend this proof to $\mathbb{R}^m$, we assume that the max-norm is reached at the component I. Then we have

$$\begin{aligned} \|\mathbf{x}\|_\infty = |x_I| = \sqrt{|x_I|^2} \leq \sqrt{\left(\sum_{i=1}^{I-1} |x_i|^2\right) + |x_I|^2 + \left(\sum_{i=I+1}^{m} |x_i|^2\right)} = \|\mathbf{x}\|_2 \\ \leq \sqrt{|x_I|^2 + \cdots + |x_I|^2} = \sqrt{m}\|\mathbf{x}\|_\infty. \end{aligned} \tag{1.13}$$

Note that these inequalities are sharp because they are attained with the following vectors

$$\| \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \|_\infty = \| \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \|_2 = 1 \quad \text{and } \| \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \|_\infty = 1, \ \| \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \|_2 = \sqrt{m}.$$

1.4 Inner product

The Euclidean inner product of two column vectors in $\mathbb{R}^m$ is defined as

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{m} x_i y_i. \tag{1.14}$$

It is a function (or map) taking two column vectors as input and returning a real number as output. Note that

$$\mathbf{x} \cdot \mathbf{x} = \sum_{i=1}^{m} x_i^2 \geq 0.$$

The Euclidean norm can be defined as the square root of the inner product of $\mathbf{x}$ with itself

$$\|\mathbf{x}\|_2 = \sqrt{\mathbf{x} \cdot \mathbf{x}} = \sqrt{\sum_{i=1}^{m} |x_i|^2}. \tag{1.15}$$

The inner product of two vectors is bounded by the product of their lengths. This inequality is the Cauchy-Schwarz inequality. It is one of the most important inequalities in mathematics. For any vectors $\mathbf{x}$ and $\mathbf{y}$ of $\mathbb{R}^m$, we write

$$|\mathbf{x} \cdot \mathbf{y}| \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2, \tag{1.16}$$

with equality if and only if $\mathbf{y}$ is proportional to $\mathbf{x}$. For example, in $\mathbb{R}^2$, we have

$$(\mathbf{x} \cdot \mathbf{y})^2 = (x_1 y_1 + x_2 y_2)^2 = x_1^2 y_1^2 + x_2^2 y_2^2 + 2 x_1 x_2 y_1 y_2.$$

Then we exploit the very useful result

$$2\alpha\beta \leq \alpha^2 + \beta^2 \quad \Longrightarrow \quad 2 x_1 x_2 y_1 y_2 \leq x_1^2 y_2^2 + x_2^2 y_1^2$$

to obtain

$$(\mathbf{x} \cdot \mathbf{y})^2 \leq (x_1^2 + x_2^2)(y_1^2 + y_2^2) = \|\mathbf{x}\|_2^2 \|\mathbf{y}\|_2^2.$$

The proof extends to $\mathbb{R}^m$.

Another useful inequality is the Hölder inequality

$$|\mathbf{x} \cdot \mathbf{y}| \leq \|\mathbf{x}\|_p \|\mathbf{y}\|_q, \tag{1.17}$$

where p and q satisfy $1/p + 1/q = 1$ and $1 \leq p, q \leq \infty$.

Inner products allow the rigorous introduction of intuitive geometrical notions such as the length of a vector or the angle between two vectors. It also provides the means of defining orthogonality between vectors (zero inner product). We will generalize its definition later in the notes.

The cosine of the angle θ between two column vectors is expressed in terms of the inner product (1.14)

$$\cos\theta = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}. \tag{1.18}$$

Note that, because of the Cauchy-Schwarz inequality (1.16), the right hand side is always in the range $[-1, 1]$. The angle θ depends on the inner product chosen. For the same two column vectors, two different inner products can give two different angle θ.

Another important concept in linear algebra is the idea of two vectors being orthogonal to one another, which is a generalization of "perpendicular". We say that $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$ are orthogonal for the inner product (1.14) when $\mathbf{x} \cdot \mathbf{y} = 0$, *i.e.* their inner product is 0. In $\mathbb{R}^2$ or $\mathbb{R}^3$, two vectors are orthogonal if and only if the lines drawn from the origin to the points with coordinates defined by $\mathbf{x}$ and $\mathbf{y}$ are perpendicular to one another.

Example 3. Consider the vectors

$$\begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 2 \\ -2 \end{bmatrix}.$$

For the Euclidian inner product, the angle between the two vectors is

$$\theta = \arccos\left(\frac{1 \times 2 + 3 \times (-2)}{\sqrt{1^2 + 3^2}\sqrt{2^2 + 2^2}}\right) = \arccos\left(\frac{-4}{\sqrt{10}\sqrt{8}}\right) = \arccos\left(\frac{-1}{\sqrt{5}}\right) = 2.0344,$$

approximately 116 degrees.

Example 4. The vectors

$$\mathbf{x} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$$

are orthogonal,

$$\mathbf{x} \cdot \mathbf{y} = 1 \times (-3) + 3 \times 1 = -3 + 3 = 0.$$

So are the vectors

$$\mathbf{x} = \begin{bmatrix} 1 \\ 3 \\ -2 \\ 4 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 4 \\ 1 \\ 4 \\ 1/4 \end{bmatrix}.$$

Indeed, we have

$$\mathbf{x} \cdot \mathbf{y} = 1 \times 4 + 3 \times 1 + (-2) \times 4 + 4\times = 4 + 3 - 8 + 1 = 0.$$

1.5 Useful commands in Matlab

Here are a few commands in MATLAB useful for this section.

- `x = [1;2;3];` defines the vector $\mathbf{x} \in \mathbb{R}^3$ with components 1, 2, and 3.
- `z = zeros(m,1)` defines the zero vector in $\mathbb{R}^m$.
- `x = ones(m,1)` defines the vector $\mathbf{x} \in \mathbb{R}^m$ with components equal to 1.

- `size(x)` returns the dimensions of `x` in a row vector `[m 1]`.
- `x = rand(m,1)` makes a vector of length m with random values, uniformly distributed between 0 and 1.
- `x = randn(m,1)` makes a vector of length m with random values, normally distributed (with mean 0 and variance 1).
- `norm(x)` computes the 2-norm of vector $\mathbf{x}$. `norm(x,inf)`, `norm(x,1)`, `norm(x,p)` computes, respectively, the ∞-norm, the 1-norm, and the p-norm.
- `dot(x,y)` computes the Euclidian inner product between vectors $\mathbf{x}$ and $\mathbf{y}$.
- `max([1;-4;3])` computes the maximum entry in the vector (here 3).
- `min([1;-4;3])` computes the minimum entry in the vector (here -4).
- The next sequence generates a vector $\mathbf{x}$ with 1000 components linearly distributed between 0 and 5

```
x = [];
for i=1:1000,
  x = [x; (i-1)*5.0/1000.0];
end
```

Whenever possible, it is recommended to declare $\mathbf{x}$ with its final size as follows

```
x = zeros(1000,1);
for i=1:1000,
  x(i) = (i-1)*5.0/1000.0;
end
```

1.6 Exercises

Exercise 5. Prove that the addition for vectors in $\mathbb{R}^m$ is associative: $(\mathbf{x}+\mathbf{y})+\mathbf{z}=\mathbf{x}+(\mathbf{y}+\mathbf{z})$.

Exercise 6. Prove that the addition of vectors in $\mathbb{R}^m$ is commutative: $\mathbf{x}+\mathbf{y}=\mathbf{y}+\mathbf{x}$.

Exercise 7. Find the unique vector $\mathbf{z}$ such that $\mathbf{x}+\mathbf{z}=\mathbf{x}$, for any vector $\mathbf{x}$. $\mathbf{z}$ is called the zero vector and, sometimes, denoted $\mathbf{0}$.

Exercise 8. Show that every vector $\mathbf{x}\in\mathbb{R}^m$ has an additive inverse $\mathbf{y}\in\mathbb{R}^m$ such that $\mathbf{x}+\mathbf{y}=\mathbf{0}$.

Exercise 9. Prove that $\forall\,\alpha,\beta\in\mathbb{R},\ \alpha(\beta\mathbf{x})=(\alpha\beta)\mathbf{x}$.

Exercise 10. Prove that $\forall\alpha\in\mathbb{R},\forall\ \mathbf{x},\ \mathbf{y}\in\mathbb{R}^m,\ \alpha(\mathbf{x}+\mathbf{y})=\alpha\mathbf{x}+\alpha\mathbf{y}$.

Exercise 11. Prove that $\forall\alpha,\ \beta\in\mathbb{R},\forall\ \mathbf{x}\in\mathbb{R}^m,\ (\alpha+\beta)\mathbf{x}=\alpha\mathbf{x}+\beta\mathbf{x}$.

Exercise 12. Show that, for every vector $\mathbf{x}\in\mathbb{R}^m$, $1\mathbf{x}=\mathbf{x}$.

Exercise 13. Check whether the following maps, defined on $\mathbb{R}^3$, are norms or not

- $\mathbf{x}\longmapsto x_1+x_2+x_3$
- $\mathbf{x}\longmapsto|x_1+x_2+x_3|$
- $\mathbf{x}\longmapsto\sqrt{x_1^4+x_2^4+x_3^4}$

Exercise 14. Prove that the following functions are norms:

1. $\forall\,\mathbf{x}=\begin{bmatrix}x_1\\x_2\end{bmatrix}\in\mathbb{R}^2,\ \|\mathbf{x}\|=|x_1+x_2|+|x_1-x_2|$
2. $\forall\,\mathbf{x}=\begin{bmatrix}x_1\\x_2\end{bmatrix}\in\mathbb{R}^2,\ \|\mathbf{x}\|=\max\left(|x_1+x_2|,|x_1-x_2|\right)$
3. $\forall\,\mathbf{x}=\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\in\mathbb{R}^3,\ \|\mathbf{x}\|=|x_1|+|x_2|+|x_3|$
4. $\forall\,\mathbf{x}\in\mathbb{R}^2,\ \|\mathbf{x}\|_2=\sqrt{|x_1|^2+|x_2|^2}$
5. $\forall\,\mathbf{x}=\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\in\mathbb{R}^3,\ \|\mathbf{x}\|=\max\left(|x_1|,|x_2|,|x_3|\right)$

Exercise 15. Prove that the p-norm satisfies the properties (1.4).

Exercise 16. Draw the unit closed ball $\{\mathbf{x}\in\mathbb{R}^2\mid\|\mathbf{x}\|\leq 1\}$ corresponding to the 4-norm and ∞-norm.

Exercise 17. Show that the function $\sqrt{|x_1|^2 + 3|x_2|^2}$ is a norm. Draw its closed unit ball.

Exercise 18. Show that the function $\sqrt{|x_1 - \sqrt{3}x_2|^2 + |\sqrt{3}x_1 + x_2|^2}$ is a norm. Draw its closed unit ball.

Exercise 19. Show that the function

$$\left\{ \begin{array}{ccc} \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} & \longmapsto & \sqrt{(x_1 - x_2)^2 + (x_1 + x_2)^2} \end{array} \right.$$

is a norm in $\mathbb{R}^2$.

Exercise 20. Prove that $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_1 \leq m\|\mathbf{x}\|_\infty$. Find vectors with equality.

Exercise 21. Prove that $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{m}\|\mathbf{x}\|_\infty$. Find vectors with equality.

Exercise 22. Prove that $\frac{1}{\sqrt{m}}\|\mathbf{x}\|_1 \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1$. Find vectors with equality.

Exercise 23. For the Euclidian norm and inner product, prove that, for any vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$,

1. $\|\mathbf{x} + \mathbf{y}\|_2^2 = \|\mathbf{x}\|_2^2 + 2\mathbf{x} \cdot \mathbf{y} + \|\mathbf{y}\|_2^2$
2. $\|\mathbf{x} - \mathbf{y}\|_2^2 + \|\mathbf{x} + \mathbf{y}\|_2^2 = 2\|\mathbf{x}\|_2^2 + 2\|\mathbf{y}\|_2^2$

   The last equality is called the parallelogram identity.

Exercise 24. Compute the angle, with the Euclidian inner product, between the following pairs of vectors

$$\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix} \right), \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right), \left(\begin{bmatrix} 1 \\ \sqrt{3} \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix} \right).$$

Exercise 25. Write a MATLAB code to compute the 1-norm, the 2-norm, the 4-norm, the 8-norm, and the max norm for the vector

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2^{1/3} \\ 3^{1/5} \\ 5^{1/7} \\ 7^{1/11} \end{bmatrix}.$$

Exercise 26. Write a MATLAB code to compute the 1-norm and the 2-norm for the following vectors

1. $\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$.

2. Reset the random generator to its 1st state. Set $\mathbf{y}$ as a random vector in $\mathbb{R}^{123}$.

Exercise 27. Write a MATLAB code to compute the angle, with the Euclidian inner product, between the following pairs of vectors

1. $\left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right)$.
2. $\left(\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ 3 \\ -2 \\ -1 \end{bmatrix} \right)$.
3. Reset the random generator to its 2nd state. Set $\mathbf{x}$ and $\mathbf{y}$ as a random vector in $\mathbb{R}^{123}$.

2 Linear Spaces

Recall that $\mathbb{R}^m$ denotes the set of all real m-vectors, *i.e.* the set of all vectors with m components, each of which is a real number. We have also defined what we mean by addition of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$: we obtain the sum by adding each component, and the sum $\mathbf{x} + \mathbf{y}$ is another vector in $\mathbb{R}^m$. We have also defined scalar multiplication: if $\mathbf{x} \in \mathbb{R}^m$ and $\alpha \in \mathbb{R}$ then the vector $\alpha\mathbf{x} \in \mathbb{R}^m$ is defined by multiplying each component of $\mathbf{x}$ by the scalar α. The set $\mathbb{R}^m$ is closed under addition and scalar multiplication. This just means that whenever these operations are applied to vectors in $\mathbb{R}^m$ we obtain another vector in the same set $\mathbb{R}^m$.

The set $\mathbb{R}^m$ is an example of linear space, which can be defined more generally as follows:

Definition 28. A real linear space, or $\mathbb{R}$-linear space, consists of a set of objects V along with two operations '$+$' (addition) and '$\cdot$' (scalar multiplication) subject to these conditions:

1. If $u, v \in V$ then $u + v \in V$ (closed under addition);
2. If $u, v \in V$ then $u + v = v + u$ (addition is commutative);
3. If $u, v, w \in V$ then $(u + v) + w = u + (v + w)$ (addition is associative);
4. There is a zero vector $\bar{0} \in V$ such that $v + \bar{0} = v$ for every $v \in V$;
5. Every $v \in V$ has an additive inverse $w \in V$ such that $v + w = \bar{0}$;
6. If $v \in V$ and $\alpha \in \mathbb{R}$ then $\alpha \cdot v \in V$ (closed under scalar multiplication);
7. If $v \in V$ and $\alpha, \beta \in \mathbb{R}$ then $(\alpha + \beta) \cdot v = \alpha \cdot v + \beta \cdot v$;
8. If $u, v \in V$ and $\alpha \in \mathbb{R}$ then $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$;
9. If $v \in V$ and $\alpha, \beta \in \mathbb{R}$ then $(\alpha\beta) \cdot v = \alpha \cdot (\beta \cdot v)$;
10. If $v \in V$ then $1 \cdot v = v$.

It is also possible to define a complex linear space or $\mathbb{C}$-linear space where the scalars are now complex.

Example 29. Verify the properties of Definition 28 for the set of column vectors $\mathbb{R}^m$.

1. For any vector $\mathbf{x}$ and any vector $\mathbf{y}$ in $\mathbb{R}^m$, we have

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \quad \Rightarrow \mathbf{x} + \mathbf{y} = \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_m + y_m \end{bmatrix}.$$

   So the vector $\mathbf{x} + \mathbf{y}$ is also a column vector with m rows and belongs to $\mathbb{R}^m$.

2. For any vector $\mathbf{x}$ and any vector $\mathbf{y}$ in $\mathbb{R}^m$, the vectors $\mathbf{x}+\mathbf{y}$ and $\mathbf{y}+\mathbf{x}$ are equal. Indeed, we have

$$\mathbf{x}+\mathbf{y}=\begin{bmatrix} x_1+y_1 \\ \vdots \\ x_m+y_m \end{bmatrix} \quad \mathbf{y}+\mathbf{x}=\begin{bmatrix} y_1+x_1 \\ \vdots \\ y_m+x_m \end{bmatrix}$$

and the m components are equal because the addition of scalar numbers is commutative.

3. The addition of column vectors is associative. Indeed, we have

$$(\mathbf{x}+\mathbf{y})+\mathbf{z}=\begin{bmatrix} (x_1+y_1)+z_1 \\ \vdots \\ (x_m+y_m)+z_m \end{bmatrix} \quad \mathbf{x}+(\mathbf{y}+\mathbf{z})=\begin{bmatrix} x_1+(y_1+z_1) \\ \vdots \\ x_m+(y_m+z_m) \end{bmatrix}.$$

The m components are equal because the addition of scalar numbers is associative.

4. The zero vector is the vector with all its m components equal to 0,

$$\mathbf{0}=\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

5. For any vector $\mathbf{x}$ in $\mathbb{R}^m$, the vector $\mathbf{y}$, defined by

$$\mathbf{x}=\begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \quad \mathbf{y}=\begin{bmatrix} -x_1 \\ \vdots \\ -x_m \end{bmatrix},$$

is such that $\mathbf{x}+\mathbf{y}=\mathbf{0}$. The vector $\mathbf{y}$ is the additive inverse of $\mathbf{x}$.

6. For any vector $\mathbf{x}$ in $\mathbb{R}^m$ and any scalar α in $\mathbb{R}$, we have

$$\mathbf{x}=\begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \quad \Rightarrow \alpha\mathbf{x}=\begin{bmatrix} \alpha x_1 \\ \vdots \\ \alpha x_m \end{bmatrix}.$$

So the vector $\alpha\mathbf{x}$ is also a column vector with m rows and belongs to $\mathbb{R}^m$.

7. For any vector $\mathbf{x}$ in $\mathbb{R}^m$ and any scalar α and β in $\mathbb{R}$, we have

$$(\alpha+\beta)\mathbf{x}=\begin{bmatrix} (\alpha+\beta)x_1 \\ \vdots \\ (\alpha+\beta)x_m \end{bmatrix}=\begin{bmatrix} \alpha x_1+\beta x_1 \\ \vdots \\ \alpha x_m+\beta x_m \end{bmatrix}=\begin{bmatrix} \alpha x_1 \\ \vdots \\ \alpha x_m \end{bmatrix}+\begin{bmatrix} \beta x_1 \\ \vdots \\ \beta x_m \end{bmatrix}=\alpha\mathbf{x}+\beta\mathbf{x}.$$

8. For any vector $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^m$ and any scalar α in $\mathbb{R}$, we have

$$\alpha(\mathbf{x}+\mathbf{y}) = \alpha \begin{bmatrix} x_1+y_1 \\ \vdots \\ x_m+y_m \end{bmatrix} = \begin{bmatrix} \alpha(x_1+y_1) \\ \vdots \\ \alpha(x_m+y_m) \end{bmatrix} = \begin{bmatrix} \alpha x_1+\alpha y_1 \\ \vdots \\ \alpha x_m+\alpha y_m \end{bmatrix} = \alpha\mathbf{x}+\alpha\mathbf{y}$$

9. For any vector $\mathbf{x}$ in $\mathbb{R}^m$ and any scalar α and β in $\mathbb{R}$, we have

$$(\alpha\beta)\mathbf{x} = \begin{bmatrix} (\alpha\beta)x_1 \\ \vdots \\ (\alpha\beta)x_m \end{bmatrix} = \begin{bmatrix} \alpha(\beta x_1) \\ \vdots \\ \alpha(\beta x_m) \end{bmatrix} = \alpha \begin{bmatrix} \beta x_1 \\ \vdots \\ \beta x_m \end{bmatrix} = \alpha(\beta\mathbf{x}).$$

10. For any vector $\mathbf{x}$ in $\mathbb{R}^m$, we have

$$1\cdot\mathbf{x} = 1\cdot\begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} 1\times x_1 \\ \vdots \\ 1\times x_m \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = \mathbf{x}.$$

So $\mathbb{R}^m$ is a real linear space.

Example 30. The set of all possible functions mapping real numbers to real numbers, $f:\mathbb{R}\longrightarrow\mathbb{R}$, is denoted $\mathcal{F}(\mathbb{R},\mathbb{R})$. We will verify that $\mathcal{F}(\mathbb{R},\mathbb{R})$ is a real linear space.

1. For any function g and h in $\mathcal{F}(\mathbb{R},\mathbb{R})$, we have

$$(g+h)(x) = g(x)+h(x).$$

So the function $f=g+h$ is a function whose input is a real number and its output is also a real number. $g+h$ belongs to $\mathcal{F}(\mathbb{R},\mathbb{R})$. For instance, if $g(x)=3x^2$ and $h(x)=\cos(x)$, then $g+h$ is the function defined by

$$(g+h)(x) = 3x^2+\cos(x), \quad \forall x\in\mathbb{R}.$$

2. For any function g and h in $\mathcal{F}(\mathbb{R},\mathbb{R})$, the functions $g+h$ and $h+g$ are equal. Indeed, we have

$$\forall x\in\mathbb{R},\ (g+h)(x) = g(x)+h(x) = h(x)+g(x) = (h+g)(x).$$

We used the fact that the addition of scalar numbers is commutative.

3. The addition of functions is associative. Indeed, we have

$$\begin{aligned}\forall x\in\mathbb{R},\ [(f+g)+h](x) &= (f+g)(x)+h(x) = f(x)+g(x)+h(x)\\ &= f(x)+(g+h)(x) = [f+(g+h)](x).\end{aligned}$$

We used the fact that the addition of scalar numbers is associative.

4. The zero function $\bar{0}$ is the function identically 0 in $\mathbb{R}$,

$$\bar{0}(x) = 0, \quad \forall x \in \mathbb{R}. \tag{2.1}$$

5. The additive inverse of a function g is the continuous function $g^{(-)}$ defined by

$$g^{(-)}(x) = -g(x), \quad \forall x \in \mathbb{R}. \tag{2.2}$$

Indeed, we have $g(x) + g^{(-)}(x) = g(x) - g(x) = 0$ for every x in $\mathbb{R}$.

6. For any function g in $\mathcal{F}(\mathbb{R}, \mathbb{R})$ and any scalar α in $\mathbb{R}$, we have

$$\forall x \in \mathbb{R}, \ (\alpha g)(x) = \alpha g(x).$$

So the function $f = \alpha g$ is a function whose input is a real number and its output is also a real number. αg belongs to $\mathcal{F}(\mathbb{R}, \mathbb{R})$. For example, if $g(x) = \cos(x)$, $5g$ is the function defined by

$$(5g)(x) = 5\cos(x), \quad \forall x \in \mathbb{R}.$$

7. For any function g in $\mathcal{F}(\mathbb{R}, \mathbb{R})$ and any scalar α and β in $\mathbb{R}$, we have

$$\forall x \in \mathbb{R}, \ [(\alpha+\beta)g](x) = (\alpha+\beta)g(x) = \alpha g(x) + \beta g(x) = (\alpha g)(x) + (\beta g)(x).$$

So the functions $(\alpha + \beta)g$ and $\alpha g + \beta g$ are equal.

8. For any function g and h in $\mathcal{F}(\mathbb{R}, \mathbb{R})$ and any scalar α in $\mathbb{R}$, we have

$$\forall x \in \mathbb{R}, \ [\alpha(g + h)](x) = \alpha g(x) + \alpha h(x) = (\alpha g)(x) + (\alpha h)(x).$$

So the functions $\alpha(g + h)$ and $\alpha g + \alpha h$ are equal.

9. For any function g in $\mathcal{F}(\mathbb{R}, \mathbb{R})$ and any scalar α and β in $\mathbb{R}$, we have

$$\forall x \in \mathbb{R}, \ [(\alpha\beta)g](x) = (\alpha\beta)g(x) = \alpha\beta g(x) = \alpha[\beta g(x)] = \alpha(\beta g)(x).$$

So the functions $(\alpha\beta)g$ and $\alpha(\beta g)$ are equal.

10. For any function g in $\mathcal{F}(\mathbb{R}, \mathbb{R})$, we have

$$\forall x \in \mathbb{R}, \ (1 \cdot g)(x) = 1 \times g(x) = g(x).$$

Example 31. The set of all continuous functions mapping real numbers to real numbers, $f : \mathbb{R} \longrightarrow \mathbb{R}$, is denoted $C^0(\mathbb{R}, \mathbb{R})$. $C^0(\mathbb{R}, \mathbb{R})$ is a real linear space.

We emphasize that the zero element takes different meanings according to the linear space V. When $V = \mathbb{R}^m$, the zero element is

$$\bar{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.3}$$

the vector with m zero components. When $V = \mathcal{C}^0(\mathbb{R}, \mathbb{R})$, the zero element $\bar{0}$ is the function identically 0 in $\mathbb{R}$,

$$\bar{0}(x) = 0, \quad \forall x \in \mathbb{R}. \tag{2.4}$$

We will mostly study linear algebra in the context of linear spaces of vectors, but the study of other linear spaces, particularly function spaces, is extremely important in many branches of mathematics and many of the ideas introduced here carry over to other linear spaces.

2.1 Subsets and subspaces

Suppose we have a linear space V and $\mathcal{S}$ is a subset of V, which just means every element of $\mathcal{S}$ is also an element of V,

$$v \in \mathcal{S} \Longrightarrow v \in V. \tag{2.5}$$

A subset might contain a finite or infinite number of elements.

Example 32. The subset,

$$\mathcal{S}_1 = \left\{ \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -2.3 \\ 7 \end{bmatrix}, \begin{bmatrix} \pi \\ 1 \end{bmatrix} \right\}, \tag{2.6}$$

is a subset of $\mathbb{R}^2$ with 3 elements.

Example 33. The subset,

$$\mathcal{S}_2 = \left\{ \mathbf{x} \in \mathbb{R}^3 : \; x_2 = x_1^2 + 3x_3 \right\}, \tag{2.7}$$

is a subset of $\mathbb{R}^3$ with an infinite number of elements, including

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -2 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix}.$$

Example 34. The subset,

$$\mathcal{S}_3 = \left\{ \mathbf{x} \in \mathbb{R}^2 : \; x_2 = 3x_1 \right\}, \tag{2.8}$$

is a subset of $\mathbb{R}^2$ with an infinite number of elements, including

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -2.3 \\ -6.9 \end{bmatrix}, \begin{bmatrix} \pi \\ 3\pi \end{bmatrix}.$$

Definition 35. If $\mathcal{S}$ is a subset of a linear space V and $\mathcal{S}$ is closed under addition and scalar multiplication, then we say that $\mathcal{S}$ is a subspace of V.

Consider the previous subsets.

- The subset $\mathcal{S}_1$ is not a subspace of $\mathbb{R}^2$ because adding two vectors from $\mathcal{S}_1$ does not give a vector in $\mathcal{S}_1$.
- The subset $\mathcal{S}_2$ is not a subspace of $\mathbb{R}^3$ because

$$2\begin{bmatrix} -2 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -4 \\ 2 \\ -2 \end{bmatrix} \notin \mathcal{S}_2.$$

- The subset $\mathcal{S}_3$ is a subspace of $\mathbb{R}^2$.

Example 36. $C^0(\mathbb{R},\mathbb{R})$ denotes the set of functions f,

$$f : \mathbb{R} \longrightarrow \mathbb{R},$$

that are continuous. For instance, $f(x) = 3x^2 + \cos(x)$ and $g(x) = |x|$ belong to $C^0(\mathbb{R},\mathbb{R})$. $C^0(\mathbb{R},\mathbb{R})$ is a subspace of $\mathcal{F}(\mathbb{R},\mathbb{R})$. Indeed, it is a subset of $\mathcal{F}(\mathbb{R},\mathbb{R})$, which is a real linear space. For any function f and g in $C^0(\mathbb{R},\mathbb{R})$, the sum function $f+g$ is continuous. So the sum function $f+g$ belongs to $C^0(\mathbb{R},\mathbb{R})$. $C^0(\mathbb{R},\mathbb{R})$ is closed for the addition. For any function f in $C^0(\mathbb{R},\mathbb{R})$ and any scalar α in $\mathbb{R}$, the function αf is continuous. So the function αf belongs to $C^0(\mathbb{R},\mathbb{R})$. $C^0(\mathbb{R},\mathbb{R})$ is closed for the scalar multiplication. Consequently, $C^0(\mathbb{R},\mathbb{R})$ is a subspace of $\mathcal{F}(\mathbb{R},\mathbb{R})$. It is also a real linear space.

Example 37. $C^1(\mathbb{R},\mathbb{R})$ denotes the set of functions f,

$$f : \mathbb{R} \longrightarrow \mathbb{R},$$

that are continuous and differentiable and whose derivative f' is also continuous. For instance, $f(x) = 3x^2 + \cos(x)$ is in $C^1(\mathbb{R},\mathbb{R})$. On the other hand, $g(x) = |x|$ belongs to $C^0(\mathbb{R},\mathbb{R})$ but not to $C^1(\mathbb{R},\mathbb{R})$. $C^1(\mathbb{R},\mathbb{R})$ is a subspace of $\mathcal{F}(\mathbb{R},\mathbb{R})$. Indeed, it is a subset of $\mathcal{F}(\mathbb{R},\mathbb{R})$, which is a real linear space. For any function f and g in $C^1(\mathbb{R},\mathbb{R})$, the sum function $f+g$ is continuous. It is also differentiable and its derivative, equal to $f'+g'$, is also continuous. So the sum function $f+g$ belongs to $C^1(\mathbb{R},\mathbb{R})$. $C^1(\mathbb{R},\mathbb{R})$ is closed for the addition. For any function f in $C^1(\mathbb{R},\mathbb{R})$ and any scalar α in $\mathbb{R}$, the function αf is continuous. It is also differentiable and its derivative, equal to $\alpha f'$, is also continuous. So the function αf belongs to $C^1(\mathbb{R},\mathbb{R})$. $C^1(\mathbb{R},\mathbb{R})$ is closed for the scalar multiplication. Consequently, $C^1(\mathbb{R},\mathbb{R})$ is a subspace of $\mathcal{F}(\mathbb{R},\mathbb{R})$. It is also a real linear space.

Example 38. $C^p(\mathbb{R},\mathbb{R})$ denotes the set of functions f,

$$f : \mathbb{R} \longrightarrow \mathbb{R},$$

that are continuous and differentiable p times and whose p-th derivative $f^{(p)}$ is also continuous. $C^p(\mathbb{R},\mathbb{R})$ is a subspace of $C^0(\mathbb{R},\mathbb{R})$.

Note the following about subspaces:

- The set $\mathcal{S} = V$ is a subspace of V ($\mathbb{R}^2$ is a subspace of $\mathbb{R}^2$). If $\mathcal{S}$ is a subspace of V that is not all of V then it is called a proper subspace of V.
- A subspace $\mathcal{S}$ of a real linear space V is also a real linear space.
- The set $\mathcal{Z} = \{\bar{0}\}$ that contains only the zero element of the linear space V is a subspace of V since $\bar{0} + \bar{0} = \bar{0}$ and $\alpha \cdot \bar{0} = \bar{0}$ so this set is closed under these operations.
- The set $\mathcal{Z} = \{\bar{0}\}$ is the only subspace of V that contains a finite number of elements (just 1 element). All other subspaces contain an infinite number of elements. Why? Because if $v \in \mathcal{S}$ then $\alpha v \in \mathcal{S}$ for any real number α (of which there are infinitely many). If $\alpha_1 v = \alpha_2 v$ then by the rules of Definition 28, we can rewrite this as $(\alpha_1 - \alpha_2)v = \bar{0}$. But this can be true only if $v = \bar{0}$ or if $\alpha_1 - \alpha_2 = 0$.

The fact that a subspace is also a real linear space suggests a technique to prove that a set U is a real linear space:

1. Find a real linear space $\mathcal{U}$ such that $U \subset \mathcal{U}$;
2. Show that U is a subspace of $\mathcal{U}$.

Example 39. Consider the set

$$\mathcal{S} = \left\{ h \in C^0(\mathbb{R}, \mathbb{R}) \,|\, h(0) = 0 \right\}.$$

We would like to check that $\mathcal{S}$ is a real linear space. Two approaches are possible:

1. Use Definition 28 and check all 10 items.
2. Find a superset U, which is a real linear space, such that $\mathcal{S} \subset U$. Show that $\mathcal{S}$ is a subspace of U by checking the properties of being closed for the addition and the multiplication (only 2 checks):
   (a) $\forall u, v \in \mathcal{S}, u + v \in \mathcal{S}$
   (b) $\forall \alpha \in \mathbb{R}, \forall u \in \mathcal{S}, \alpha u \in \mathcal{S}$

To prove that $\mathcal{S}$ is a real linear space, we will use the second approach. We need to find a superset. The definition of $\mathcal{S}$ suggests $C^0(\mathbb{R}, \mathbb{R})$ as superset. Indeed, every function in $\mathcal{S}$ is a continuous function. We know that $C^0(\mathbb{R}, \mathbb{R})$ is a real linear space. So we just need to check the properties of being closed.

Let f and g be two functions in $\mathcal{S}$. The sum $f + g$ is a continuous function because f and g are continuous. To check whether $f + g$ belongs to $\mathcal{S}$, we need also to compute the value of $f + g$ at 0:

$$(f + g)(0) = f(0) + g(0) = 0 + 0 = 0.$$

So $f + g$ belongs to $\mathcal{S}$.

Let f be a functions in $\mathcal{S}$ and $\alpha \in \mathbb{R}$. The product αf is a continuous function because f is continuous. To check whether αf belongs to $\mathcal{S}$, we need also to compute the value of αf at 0:

$$(\alpha f)(0) = \alpha f(0) = \alpha 0 = 0.$$

So αf belongs to $\mathcal{S}$.

We conclude that $\mathcal{S}$ is a subspace of $C^0(\mathbb{R}, \mathbb{R})$ and it is also a real linear space.

2.2 Linear dependence and independence

If $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$ are any two vectors and $\alpha = \beta = 0$, then

$$\alpha\mathbf{x} + \beta\mathbf{y} = 0\mathbf{x} + 0\mathbf{y} = \mathbf{0} + \mathbf{0} = \mathbf{0}.$$

So this trivial linear combination of $\mathbf{x}$ and $\mathbf{y}$ is always the zero vector.

In a real linear space V, two elements $u, v \in V$ are said to be linearly dependent if there is some nontrivial linear combination of u and v that gives $\bar{0}$, *i.e.* if there are scalars $\alpha, \beta \in \mathbb{R}$ that are not both equal to zero but for which $\alpha u + \beta v = \bar{0}$. In general, two elements in V are linearly dependent if (and only if) one is a scalar multiple of the other, for example if $v = \gamma u$ for some scalar γ. Since then $\gamma u - v = \bar{0}$ (or any other linear combination with $\beta \neq 0$ and $\alpha = -\gamma\beta$ gives the zero vector).

Two elements (or objects) in V are said to be linearly independent when there is no nontrivial linear combination of u and v that gives $\bar{0}$. In other words, they are linearly independent if the equation

$$\alpha u + \beta v = \bar{0} \tag{2.9}$$

has only the trivial solution $\alpha = \beta = 0$. Two vectors are linearly independent when one is not a scalar multiple of the other.

Example 40. The vectors

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

are linearly independent since neither one is a scalar multiple of the other. Another way to see this is that the equation $\alpha u + \beta v = \bar{0}$ is

$$\begin{bmatrix} \alpha + 3\beta \\ 2\alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The second component is zero only if $\alpha = 0$. But then the first component becomes 3β, which is zero only if $\beta = 0$. So (2.9) is satisfied only when $\alpha = \beta = 0$.

Example 41. The polynomials p and q, defined by

$$p(x) = 1, \quad q(x) = x, \quad \forall x \in \mathbb{R},$$

are linearly independent. Indeed, the equation $\alpha p + \beta q = \bar{0}$ becomes

$$\alpha + \beta x = 0, \quad \forall x \in \mathbb{R}.$$

Taking $x = 0$ gives that α must be 0. Then any nonzero value of x implies that $\beta = 0$. So (2.9) is satisfied only when $\alpha = \beta = 0$.

The idea of linear dependence and independence can be extended to sets of more than 2 elements.

Definition 42. The set of r elements $u^{(1)}, u^{(2)}, \cdots, u^{(r)} \in V$ is linearly independent if the equation

$$\alpha_1 u^{(1)} + \alpha_2 u^{(2)} + \cdots + \alpha_r u^{(r)} = \bar{0} \tag{2.10}$$

has only the trivial solution $\alpha_1 = \alpha_2 = \cdots = \alpha_r = 0$, *i.e.* if every nontrivial linear combination of the elements is nonzero. When the set is not linearly independent, it is said to be linearly dependent.

Example 43. The vectors

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

are linearly dependent. Consider a linear combination resulting in the zero vector

$$\alpha\mathbf{x} + \beta\mathbf{y} + \gamma\mathbf{z} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Then we have

$$\begin{bmatrix} \alpha + 3\beta + \gamma \\ 2\alpha + \gamma \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Identifying each entry results in the following system of equations

$$\begin{cases} \alpha + 3\beta + \gamma &= 0 \\ 2\alpha + \gamma &= 0 \end{cases}$$

These equations imply that $\gamma = -2\alpha$ and $\beta = \alpha/3$. However, α remains arbitrary. For example, a linear combination with $\alpha = 3$, $\beta = 1$, and $\gamma = -6$ results in the zero vector. This non-trivial combination implies that the 3 vectors are linearly dependent.

Example 44. The vectors,

$$\mathbf{x}^{(1)} = \begin{bmatrix} 7 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{x}^{(2)} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{x}^{(3)} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix},$$

are linearly independent. Indeeed, $\alpha_1\mathbf{x}^{(1)} + \alpha_2\mathbf{x}^{(2)} + \alpha_3\mathbf{x}^{(3)} = \mathbf{0}$ if and only if $\alpha_1 = \alpha_2 = \alpha_3 = 0$. Let us prove this result. If $\alpha_1 = \alpha_2 = \alpha_3 = 0$, then $\alpha_1\mathbf{x}^{(1)} + \alpha_2\mathbf{x}^{(2)} + \alpha_3\mathbf{x}^{(3)} = \mathbf{0}$. If a linear combination of $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, and $\mathbf{x}^{(3)}$is zero, then we have

$$\alpha_1\mathbf{x}^{(1)} + \alpha_2\mathbf{x}^{(2)} + \alpha_3\mathbf{x}^{(3)} = \begin{bmatrix} 7\alpha_1 + \alpha_2 + 3\alpha_3 \\ 2\alpha_2 + 4\alpha_3 \\ 5\alpha_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

By matching the third component, we get that α_3 has to be 0. Plugging this value into the second component, we obtain that α_2 is also 0. Then the first component implies that α_1 is 0. A linear combination of $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, and $\mathbf{x}^{(3)}$is zero if and only if $\alpha_1 = \alpha_2 = \alpha_3 = 0$.

Example 45. The functions 1, x, and x^2 are linearly independent. Indeed, consider the linear combination

$$\alpha + \beta x + \gamma x^2 = 0, \quad \forall x \in \mathbb{R} \tag{2.11}$$

Taking $x = 0$ implies that α must be 0. Then taking $x = 1$ and $x = -1$ gives

$$\begin{aligned} \beta + \gamma &= 0 \\ -\beta + \gamma &= 0 \end{aligned}$$

which implies that $\beta = \gamma = 0$. Summing the two equations results in $\beta = 0$. Plugging $\alpha = \beta = 0$ in any equation gives that $\gamma = 0$. So the linear combination (2.11) is zero if and only if $\alpha = \beta = \gamma = 0$.

Example 46. We will check the linear dependency of the functions 1, e^x, and e^{-x}. Consider a linear combination that is equal to the zero function:

$$\alpha + \beta e^x + \gamma e^{-x} = 0, \quad \forall x \in \mathbb{R}.$$

Differentiating this relationship, we have also

$$\beta e^x - \gamma e^{-x} = 0, \quad \forall x \in \mathbb{R}.$$

Taking a second derivative, we get

$$\beta e^x + \gamma e^{-x} = 0, \quad \forall x \in \mathbb{R}.$$

Summing the last two equations, we obtain

$$2\beta e^x = 0, \quad \forall x \in \mathbb{R},$$

which implies that $\beta = 0$. Plugging this value into the equation after one differentiation, we have

$$-\gamma e^{-x} = 0, \quad \forall x \in \mathbb{R}$$

and $\gamma = 0$. Finally, the first relation gives that $\alpha = 0$. Consequently, the functions 1, e^x, and e^{-x} are linearly independent.

2.3 Span of a set of elements

Let V denote a real linear space and $u^{(1)}, \cdots, u^{(r)} \in V$ be a set of r elements. Then the span of this set of elements is the space of all linear combinations of these elements,

$$\text{span}(u^{(1)}, \cdots, u^{(r)}) = \left\{ \alpha_1 u^{(1)} + \cdots + \alpha_r u^{(r)}; \alpha_1, \cdots, \alpha_r \in \mathbb{R} \right\}. \tag{2.12}$$

This is a subspace of V. Since any linear combination of elements in this set is again a linear combination of $u^{(1)}, \cdots, u^{(r)} \in V$.

Example 47. The subspace $\mathcal{S}_3$, given by (2.8), can be written as

$$\mathcal{S}_3 = \text{span} \left\{ \begin{bmatrix} 1 \\ 3 \end{bmatrix} \right\}.$$

Example 48. The space

$$\mathcal{S} = \text{span} \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

is all of $\mathbb{R}^2$, since any vector $\mathbf{x} \in \mathbb{R}^2$ can be written as a linear combination of these two vectors:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Example 49. The space

$$\mathcal{S} = \text{span} \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix} \right)$$

is all of $\mathbb{R}^2$, since any vector $\mathbf{x} \in \mathbb{R}^2$ can be written as a linear combination of these three vectors:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 0 \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

Actually, in this case, there are infinitely many different ways to write an arbitrary vector $\mathbf{x} \in \mathbb{R}^2$ as a linear combination of these three vectors. For example, we could write it as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = -x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + (x_2 - 3x_1) \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 2x_1 \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

Example 50. The set $\mathcal{P}_2$ of real polynomials of degree at most 2 is a real linear space. Any polynomial in $\mathcal{P}_2$ is written as

$$p(x) = \alpha + \beta x + \gamma x^2.$$

It is a linear combination of functions 1, x, and x^2. So we have

$$\mathcal{P}_2 = \text{span} \left(1, x, x^2\right).$$

2.4 Basis

Consider V a real linear space and $u^{(1)},\ \cdots,\ u^{(r)} \in V$ be a set of r elements. Then the span of this set of elements,

$$\mathcal{S} = \text{span}(u^{(1)}, \cdots, u^{(r)}) = \left\{ \alpha_1 u^{(1)} + \cdots + \alpha_r u^{(r)}; \alpha_1, \cdots, \alpha_r \in \mathbb{R} \right\}, \quad (2.13)$$

defines a subspace of V.

Example 51. In fact it can be shown that any subspace $\mathcal{S}$ of $\mathbb{R}^m$ has this form — it is the span of some set of vectors.

A minimal set of elements that define a space is called a basis for the space. What do we mean by minimal? Consider the following example. The subspace $\mathcal{S}_3$, given by (2.8), can be written as

$$\mathcal{S}_3 = \text{span}\left\{ \begin{bmatrix} 1 \\ 3 \end{bmatrix} \right\} = \text{span}\left\{ \begin{bmatrix} 2 \\ 6 \end{bmatrix} \right\} = \text{span}\left\{ \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \end{bmatrix} \right\}.$$

In the latter case, note that the last two vectors are linearly dependent. Clearly, we require at least one vector to define this particular space, but specifying two vectors is redundant. We say that either one of these vectors alone is a basis for this particular space. More generally we make the following definition to make this idea precise:

Definition 52. If $\mathcal{S}$ is a subspace of a linear space V, then the set of elements $u^{(1)},\ \cdots,\ u^{(r)} \in V$ form a basis for $\mathcal{S}$ if

$$\mathcal{S} = \text{span}(u^{(1)},\ \cdots,\ u^{(r)}), \quad (2.14a)$$

$$u^{(1)},\ \cdots,\ u^{(r)} \text{are linearly independent.} \quad (2.14b)$$

If a set of elements spanning the space are not linearly independent, then we can find a basis consisting of fewer elements.

Example 53. The vectors

$$\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

form a basis of $\mathbb{R}^2$. Indeed they span $\mathbb{R}^2$ and they are linearly independent.

Example 54. The space $\mathcal{P}_2$ is spanned by the functions 1, x, and x^2. These functions are linearly independent. Consequently, $(1, x, x^2)$ form a basis for $\mathcal{P}_2$.

2.5 Dimension of a space

There are typically infinitely many choices of basis for a given space $\mathcal{S}$. However, it can be shown that all choices of basis consist of the same number of elements (or objects) in $\mathcal{S}$.

Definition 55. The number of basis elements needed to define a space is called the dimension of the space.

Example 56. The space $\mathcal{S}_3$, given by (2.8), is a 1-dimensional subspace of $\mathbb{R}^2$. The single vector

$$\begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

is one possible basis for this space.

Example 57. The space $\mathcal{S}_0 = \{\bar{0}\}$, consisting only of the zero vector, is a special case. We say it is a 0-dimensional space. This is the only linear space that consists of a single vector and the only linear space that has no basis.

Example 58. The space $\mathcal{P}_2$, consisting of polynomials of degree at most 2, is of dimension 3. Indeed, $(1, x, x^2)$ form a basis for $\mathcal{P}_2$.

Example 59. The vector space $\mathbb{R}^m$ has dimension m. The "standard basis" consists of the "unit vectors",

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \ \cdots, \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}. \tag{2.15}$$

There are infinitely many other bases: any set of m linearly independent vectors will do.

Example 60. The set of complex values $\mathbb{C}$ is a real linear space of dimension 2. It is also a complex linear space of dimension 1.

Finally, we have the following important result comparing dimensions between a real linear space and one of its subspaces.

Proposition 61. *Let V be a real linear space and $\mathcal{S} \subset V$ be a subspace. Then we have*

$$\dim\mathcal{S} \leq \dim V \tag{2.16}$$

where dim *denotes the dimension of a real linear space.*

We will look for the dimension and a basis for a couple of spaces. Note that an infinite number of bases exist. However, all the bases are composed of the same number of elements. For each space, a systematic approach to find a basis is defined in the following steps.

1. Check that the space $\mathcal{S}$ is indeed a real linear space. For example, verify that it is a subspace of a bigger real linear space.

2. Write a general formula characterizing any element of $\mathcal{S}$ (an element can be a function, a vector, ...).

3. Identify the independent parameters in the formula.
4. Write a new formula characterizing any element of $\mathcal{S}$ as a linear combination of elements where each parameter is multiplying one element. This formula should indicate that these elements span the space $\mathcal{S}$.
5. Check whether these particular elements are linearly independent. If they are linearly independent, these elements form a basis for $\mathcal{S}$. If not, look for a subset of these elements that still spans $\mathcal{S}$ and that contains linearly independent elements.

Next, we apply these steps to the different examples.

Example 62. Consider the set $\mathcal{S} = \left\{\mathbf{u} \in \mathbb{R}^3 \,|\, u_3 = 0\right\}$. $\mathcal{S}$ is composed of vectors with 3 rows, whose last component, u_3, is equal to 0. $\mathcal{S}$ is a real linear space. Indeed it is a subset of $\mathbb{R}^3$, which is a real linear space. $\mathcal{S}$ is closed for the addition. For any vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathcal{S}$, we have

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ 0 \end{bmatrix} \Rightarrow \mathbf{x} + \mathbf{y} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ 0 \end{bmatrix}.$$

The vector $\mathbf{x} + \mathbf{y}$ is a vector with 3 rows and its last component is equal to 0. So $\mathbf{x} + \mathbf{y}$ belongs to $\mathcal{S}$ and $\mathcal{S}$ is closed for the addition. It is also closed for the scalar multiplication. For any vector $\mathbf{x}$ in $\mathcal{S}$ and any scalar α in $\mathbb{R}$, we have

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} \Rightarrow \alpha\mathbf{x} = \begin{bmatrix} \alpha x_1 \\ \alpha x_2 \\ 0 \end{bmatrix}.$$

The vector $\alpha\mathbf{x}$ is a vector with 3 rows and its last component is equal to 0. So $\alpha\mathbf{x}$ belongs to $\mathcal{S}$ and $\mathcal{S}$ is closed for the scalar multiplication. So $\mathcal{S}$ is a subspace of $\mathbb{R}^3$ and a real linear space. To find a basis for $\mathcal{S}$ and the dimension, we write a general formula for any vector in $\mathcal{S}$

$$\mathbf{x} \in \mathcal{S} \iff \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}.$$

This general formula has two parameters x_1 and x_2. So we can write

$$\mathbf{x} \in \mathcal{S} \iff \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

So any vector in $\mathcal{S}$ is a linear combination of the two vectors $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, which implies that

$$\mathcal{S} = \text{span}\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\right).$$

To obtain a basis, we need to check whether these two vectors are linearly independent. Consider a linear combination equal to the vector $\mathbf{0}$

$$\alpha \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Identifying the entries, we obtain that $\alpha = \beta = 0$. The vectors are linearly independent and they span $\mathcal{S}$. So they form a basis for $\mathcal{S}$. The dimension of $\mathcal{S}$ is 2.

Example 63. Check whether the set $\left\{ f \in C^1(\mathbb{R}, \mathbb{R}) \mid f'(x) = f(x) \right\}$ is a real linear space. Determine a basis and the dimension. Denote $\mathcal{S}$ the set to study. We notice that $\mathcal{S}$ is a subset of $C^1(\mathbb{R}, \mathbb{R})$, which is a real linear space. So we will show that $\mathcal{S}$ is a subspace of $C^1(\mathbb{R}, \mathbb{R})$. This will prove also that $\mathcal{S}$ is a real linear space. So we need to check whether $\mathcal{S}$ is closed for the addition and closed for the scalar multiplication. Consider any function f and g in $\mathcal{S}$, then the function $f + g$ is a function in $C^1(\mathbb{R}, \mathbb{R})$ because $C^1(\mathbb{R}, \mathbb{R})$ is a real linear space. To check whether $f + g$ belongs to $\mathcal{S}$, we have to compute the derivative to see if it is equal to $f + g$:

$$(f + g)'(t) = f'(t) + g'(t) = f(t) + g(t) = (f(t) + g(t)) = (f + g)(t)$$

So $f + g$ belongs to $\mathcal{S}$. Finally, we need to check whether $\mathcal{S}$ is closed for the multiplication. Consider any function f in $\mathcal{S}$ and any real number α, then the function αf is a function in $C^1(\mathbb{R}, \mathbb{R})$ because $C^1(\mathbb{R}, \mathbb{R})$ is a real linear space. To check whether αf belongs to $\mathcal{S}$, we have to compute the derivative to see if it is equal to αf:

$$(\alpha f)'(t) = \alpha f'(t) = \alpha(f(t)) = \alpha f(t) = (\alpha f)(t).$$

So αf belongs to $\mathcal{S}$. We conclude that $\mathcal{S}$ is a subspace of $C^1(\mathbb{R}, \mathbb{R})$ and it is also a real linear space. Any function in $\mathcal{S}$ is solution to the equation $f'(t) = f(t)$. A general solution is $f(t) = \alpha e^t$, for any $\alpha \in \mathbb{R}$. No other solution exists. Any solution is proportional to e^t. So we have $\mathcal{S} = \text{span}(e^t)$. A non-zero function is necessarily linearly independent. So (e^t) is a basis for $\mathcal{S}$. The dimension of $\mathcal{S}$ is 1.

Example 64. Consider the set $\mathcal{S} = \{p \in \mathcal{P}_3 \,|\, p(0) = p(1) = p(2) = 0\}$. $\mathcal{S}$ is composed of polynomials of degree at most 3 that take the value 0 when evaluated at $x = 0$, $x = 1$, and $x = 2$. $\mathcal{S}$ is a real linear space. Indeed it is a subset of $\mathcal{P}_3$, which is a real linear space. $\mathcal{S}$ is closed for the addition because, for any polynomials p and q in $\mathcal{S}$, $p+q$ is a polynomial in $\mathcal{P}_3$. When evaluated at 0 (or 1 or 2), $p + q$ is equal to zero because $p(0) = q(0) = 0$ (or $p(1) = q(1) = 0$ or $p(2) = q(2) = 0$). So $\mathcal{S}$ is closed for the addition. It is also closed for the multiplication because the values at 0 of αp is equal to $\alpha p(0) = 0$ (the same thing holds for $x = 1$ and $x = 2$). So $\mathcal{S}$ is a real linear space. We can write a general formula for any polynomial in $\mathcal{S}$

$$p(x) = \alpha + \beta x + \gamma x^2 + \delta x^3.$$

Evaluating the polynomial at 0, 1, and 2, we get the relations

$$\begin{array}{lrcl} (x=0) & \alpha & = & 0 \\ (x=1) & \alpha+\beta+\gamma+\delta & = & 0 \\ (x=2) & \alpha+2\beta+4\gamma+8\delta & = & 0 \end{array}$$

Substracting two times the next-to-last equation from the last equation, we get

$$2\beta+4\gamma+8\delta-(2\beta+2\gamma+2\delta)=2\gamma+6\delta=0 \quad \Longrightarrow \quad \gamma=-3\delta.$$

Then we get for $\beta=-\gamma-\delta=3\delta-\delta=2\delta$. So any polynomial in $\mathcal{S}$ has the form

$$p(x)=\delta(2x-3x^2+x^3).$$

The space $\mathcal{S}$ is spanned by the polynomial $2x-3x^2+x^3$, which is non-zero. The polynomial $2x-3x^2+x^3$ forms a basis. The dimension of $\mathcal{S}$ is 1.

2.6 Exercises

Exercise 65. Verify that $\mathbb{R}^m$ is a $\mathbb{Q}$-linear space, where $\mathbb{Q}$ is the set of signed rational numbers.

Exercise 66. Verify that $\mathbb{C}$ is a real linear space.

Exercise 67. Verify that $\mathbb{C}$ is a complex linear space.

Exercise 68. Show that the set $\mathcal{P}_n$ of all polynomials with real coefficients of degree n is a real linear space. What is the zero vector?

Exercise 69. Show that the set $\mathcal{P}_n$ of all polynomials with real coefficients of degree n is a subspace of $C^0(\mathbb{R}, \mathbb{R})$.

Exercise 70. Check the linear dependency of the vectors

$$\mathbf{x}^{(1)} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \qquad \mathbf{x}^{(2)} = \begin{bmatrix} -2.1 \\ -4.2 \\ -6.3 \end{bmatrix}.$$

Exercise 71. Check the linear dependency of the vectors

$$\mathbf{x}^{(1)} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{x}^{(2)} = \begin{bmatrix} -2 \\ 1 \\ 1.5 \end{bmatrix}, \quad \mathbf{x}^{(3)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Exercise 72. Which of the following sets are real linear spaces (where addition and scalar multiplication are defined in the usual way for these sort of objects)? Justify your answers with an argument of why it is closed under addition and scalar multiplication or a counterexample showing it is not.

1. $\{f \in C^0(\mathbb{R}, \mathbb{R}) : \int_0^2 f(t)\, dt = 0\}$, for example $f(t) = \sin(\pi t)$ and $g(t) = t - 1$ are in this set.
2. $\{f \in C^0(\mathbb{R}, \mathbb{R}) : \int_0^2 f(t)\, dt = 2\}$, for example $f(t) = 1 + \sin(\pi t)$ and $g(t) = t$ are in this set.
3. $\{f \in C^1(\mathbb{R}, \mathbb{R}) : f'(t) = 2f(t)\ \forall t \in \mathbb{R}\}$, for example $f(t) = 3e^{2t}$ is in this set.
4. $\{f \in C^1(\mathbb{R}, \mathbb{R}) : f'(t) = -3f(t)\ \forall t \in \mathbb{R}\}$.
5. $\{\mathbf{u} \in \mathbb{R}^3 : u_1 u_2 u_3 = 0\}$, for example $\mathbf{u} = [1; 0; 4]$ is in this set.

Exercise 73. Show that the functions 1, $\cos(\pi x)$, and $\sin(\pi x)$ are linearly independent on $[-1, 1]$.

Exercise 74. Show that the functions 1, $\cosh t$, and $\sinh t$ are linearly independent when t belongs to $\mathbb{R}$.

Exercise 75. What is the dimension of $\mathcal{P}_n$ the real linear space of polynomials with real coefficients of degree less or equal to n?

Exercise 76. What is the dimension of the real linear space $\mathbb{R}$? Exhibit a basis. What is the dimension of $\mathbb{R}$ as a $\mathbb{Q}$-linear space?

Exercise 77. (from P. Olver & C. Shakiban) Determine whether

1. $\begin{pmatrix} 1 \\ -2 \\ -3 \end{pmatrix}$ is in the span of $\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$.
2. $\begin{pmatrix} 1 \\ -2 \\ -1 \end{pmatrix}$ is in the span of $\begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$, $\begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}$, and $\begin{pmatrix} 0 \\ 3 \\ 4 \end{pmatrix}$.
3. $\begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$ is in the span of $\begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ -1 \\ 3 \end{pmatrix}$, and $\begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$.

Exercise 78. (from P. Olver & C. Shakiban) Which of the following functions lies in the subspace spanned by the functions $1,\ x,\ \sin x,\ \sin^2 x$:

1. $3 - 5x$
2. $\sin x - 2\cos x$
3. $x \sin x$
4. e^x
5. $x^2 + \sin^2 x$
6. $\cos^2 x$

Exercise 79. (from P. Olver & C. Shakiban) Which of the following sets of vectors form a basis of $\mathbb{R}^2$?

1. $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$
2. $\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
3. $\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}$
4. $\begin{pmatrix} 1 \\ 11 \end{pmatrix}, \begin{pmatrix} 11 \\ 1 \end{pmatrix}$
5. $\begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -2 \end{pmatrix}$

6. $\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 2 \\ -1 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}$

7. $\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 4 \\ 4 \end{pmatrix}$

Exercise 80. Show that the following sets are real linear spaces. Determine a basis and the dimension of the space:

1. The set of all vectors of the form $\begin{bmatrix} r-s \\ r+2s \\ -s \end{bmatrix}$, for $r, s \in \mathbb{R}$. (Hint: Dimension 2)
2. $\mathcal{S} = \left\{ f \in C^2(\mathbb{R}, \mathbb{R}) \,:\, f''(t) = 0,\ \forall t \in \mathbb{R} \right\}$. (Hint: Dimension 2)
3. $\mathcal{S} = \{ p \in \mathcal{P}_2 \,:\, p(0) = p(1) = 0 \}$. (Hint: Dimension 1)
4. $\mathcal{S} = \{ p \in \mathcal{P}_3 \,:\, p(0) = p'(0) = 0 \}$. (Hint: Dimension 2)

Exercise 81. Explain why the following sets are not a real linear space

1. $\mathbb{Z} = \{0, 1, -1, 2, -2, \cdots\} \subset \mathbb{R}$, the set of all signed integers.
2. The set of all vectors of the form $\begin{bmatrix} r-s \\ r+2s \\ 1 \end{bmatrix}$, for $r, s \in \mathbb{R}$
3. The set of all vectors $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, such that $x_1 + x_2 x_3 = 1$
4. The set of all non-negative functions: $f(x) \geq 0$
5. $\mathcal{S} = \left\{ f \in C^1(\mathbb{R}, \mathbb{R}) \,:\, f''(t) = 2,\ \forall t \in \mathbb{R} \right\}$

3 Linear Functions

First, recall what we mean by the notation

$$f : U \to V. \tag{3.1}$$

The function f takes an element of U as input. It is defined for any element of U. The function value $f(u)$ is an element of V.

We are now ready to define what we mean by a linear function.

Definition 82. Consider U and V two real linear spaces. The function $f : U \to V$ is a linear function if both of the following conditions are satisfied:

$$\forall u^{(1)}, u^{(2)} \in U,\ f(u^{(1)} + u^{(2)}) = f(u^{(1)}) + f(u^{(2)}) \tag{3.2a}$$

$$\forall u \in U \text{ and } \alpha \in \mathbb{R},\ f(\alpha u) = \alpha f(u) \tag{3.2b}$$

When U and V are two complex linear spaces, condition (3.2b) is modified such that α belongs to $\mathbb{C}$. Note that $f(\bar{0}) = \bar{0}$, by taking $\alpha = 0$.

3.1 Linear functions from $\mathbb{R}$ to $\mathbb{R}$

The conditions (3.2) are very restrictive. The *only* functions satisfying these conditions are functions of the form

$$f(x) = ax \tag{3.3}$$

where a is some fixed real number. Indeed, we have

$$f(x) = f(x \cdot 1) = xf(1) \tag{3.4}$$

and

$$f(x+y) = f(x)+f(y) = f(x\cdot 1)+f(y\cdot 1) = xf(1)+yf(1) = (x+y)f(1). \tag{3.5}$$

The graph of such a function is simply a line through the origin with slope a, as illustrated in Figure 3.1 for two choices of a. The fact that the graph is a line helps us remember why these functions are called linear.

Now consider the function $g(x) = 6x-3$, whose graph is shown in Figure 3.2. This graph is also a straight line but this is not a linear function according to the strict terms of Definition 82. It cannot be since $g(0) \neq 0$ and we can also easily check that $g(1+2) = 15$ while $g(1)+g(2) = 12$, for example. The function $g(x)$ is properly called an affine function although people are often imprecise and call it linear. It really consists of a linear function $6x$ plus a translation (shifting each point downwards by 3), and this is what is meant by an affine function more generally: a linear function shifted a constant (in this case -3).

Another way to define an affine function is to say that $g(x)$ is affine if $g(x) - g(y)$ is a linear function of $x - y$, or equivalently if, for any fixed point x_0 the function $f(s) = g(x_0 + s) - g(x_0)$ is a linear function of s. You can check

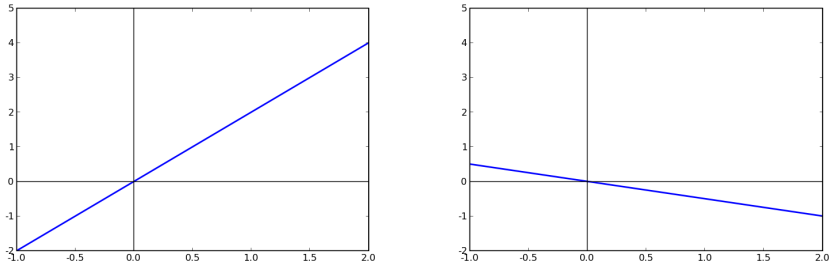


Figure 3.1: (Left) Graph of $f(x) = 2x$. (Right) Graph of $f(x) = -0.5x$.

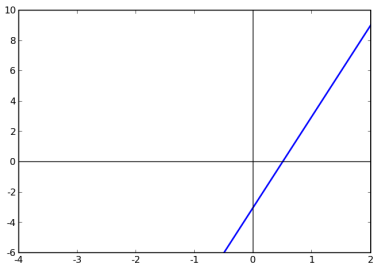


Figure 3.2: Graph of the affine function $g(x) = 6x - 3$.

that this is true for any function of the form $g(x) = ax + b$, and these are the only affine functions for the case we are currently considering, the simple case of a function mapping $\mathbb{R}$ to $\mathbb{R}$. (Soon these ideas will be generalized to more interesting situations, so make sure you understand these basic ideas even if they seem trivial now!)

3.2 Linear functions from $\mathbb{R}$ to $\mathbb{R}^m$

We might have a situation where there are two different output values $f_1(x)$ and $f_2(x)$ that depend on all the same input value x. We could then use the symbol $\mathbf{f}(x)$ to denote the vector

$$\mathbf{f}(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \tag{3.6}$$

containing these two output values. If the outputs are real numbers and are defined for any real number x, then we would say that $\mathbf{f}$ maps $\mathbb{R}$ to $\mathbb{R}^2$.

This function is said to be linear if *both* real functions $f_1(x)$ and $f_2(x)$ are linear functions in the sense of Definition 82. From what we noticed in the previous section, we see that a function $\mathbf{f} : \mathbb{R} \to \mathbb{R}^2$ is linear only if it has the

form

$$\mathbf{f}(x) = \begin{bmatrix} a_1 x \\ a_2 x \end{bmatrix} \tag{3.7}$$

where the values a_1 and a_2 are two scalar constants.

More generally, a function $\mathbf{f} : \mathbb{R} \to \mathbb{R}^m$ is linear if and only if it has the form

$$\mathbf{f}(x) = \begin{bmatrix} a_1 x \\ \vdots \\ a_m x \end{bmatrix} \tag{3.8}$$

where the values $a_1, \cdots, a_m$ are m scalar constants.

3.3 Linear functions from $\mathbb{R}^n$ to $\mathbb{R}$

Suppose f is a function that depends on several input values, say x_1, x_2, and x_3. To make the notation simple, we can still talk about the function $f(\mathbf{x})$ if we now think of $\mathbf{x}$ as the vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{3.9}$$

Using the standard unit basis (2.15), we write a linear combination in terms of the unit basis vectors

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ x_3 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{3.10}$$

It is easy to show

$$f(\mathbf{x}) = x_1 f\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\right) + x_2 f\left(\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\right) + x_3 f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right). \tag{3.11}$$

An essential property of a linear function is the following: if we know the values of

$$f\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\right), f\left(\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\right), \text{ and } f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right), \tag{3.12}$$

then we can easily find the value of $f(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^3$. Indeed, any vector of $\mathbb{R}^3$ is a linear combination of basis vectors

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \text{ and } \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{3.13}$$

Taking linear combinations of vectors is a fundamental operation in linear algebra.

It turns out that if $f: \mathbb{R}^n \to \mathbb{R}$ is a linear function, then all we need to do is evaluate the function f for some well chosen set of n linearly independent vectors and we can evaluate $f(\mathbf{x})$ for any $\mathbf{x}$ in all of $\mathbb{R}^n$. This is one of the reasons why linear problems and linear equations are so important in applications.

You might guess that every linear function mapping $\mathbb{R}^3$ to $\mathbb{R}$ must have the form

$$f(\mathbf{x}) = a_1x_1 + a_2x_2 + a_3x_3 \tag{3.14}$$

for some constant real numbers a_1, a_2, and a_3. You would be right! More generally, a function $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}$ is linear if and only if it has the form

$$f(\mathbf{x}) = a_1x_1 + a_2x_2 + \cdots + a_nx_n \tag{3.15}$$

where the values $a_1, \cdots, a_n$ are n scalar constants.

3.4 Linear functions from $\mathbb{R}^n$ to $\mathbb{R}^m$

Now suppose that the function $\mathbf{f}$ is a vector with m components that depends on n input values. We can write

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}. \tag{3.16}$$

According to Definition 82, $\mathbf{f}$ is a linear function if and only if the m components $f_1, \cdots, f_m$ are linear functions of $\mathbf{x}$.

From what we noticed in the previous sections, the ith function $f_i(\mathbf{x})$ is then given by

$$f_i(\mathbf{x}) = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = \sum_{j=1}^{n} a_{ij}x_j \tag{3.17}$$

for $i = 1,\ 2,\ \ldots,\ m$.

We often work with linear functions with many inputs and outputs and so it is nice to simplify the notation for describing these functions. This was recognized many years ago and so the notation of a matrix was invented. The function is uniquely determined by mn numbers that are naturally arranged in a matrix with m rows and n columns:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}. \tag{3.18}$$

A common notation is

$$\mathbf{A} = (a_{ij})_{i=1,\cdots,m;j=1,\cdots,n} = (a_{ij})_{m\times n} \tag{3.19}$$

For shorthand, we might write

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}, \tag{3.20}$$

where $\mathbf{x}$ is the vector of inputs with n components. This notation suggests that we multiply the $m \times n$ matrix $\mathbf{A}$ by the vector $\mathbf{x}$ to obtain the vector $\mathbf{f}(\mathbf{x})$.

Definition 83. The set of all real matrices with m rows and n columns is denoted $\mathbb{R}^{m\times n}$.

We define the concept of matrix-vector multiplication so that this is correct:

Definition 84. If $\mathbf{A}$ is an $m \times n$ matrix of the form (3.18) and $\mathbf{x} \in \mathbb{R}^n$ is an n-vector, then the product $\mathbf{b} = \mathbf{A}\mathbf{x}$ is an m-vector ($\mathbf{b} \in \mathbb{R}^m$) and the ith component of $\mathbf{b}$ is

$$b_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = \sum_{j=1}^{n} a_{ij}x_j. \tag{3.21}$$

The matrix-vector multiplication can also be displayed as follows:

$$\mathbf{b} = \begin{bmatrix} & & & \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \\ & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \ldots + x_n\mathbf{a}_n$$

where $\mathbf{b}$ is expressed as a linear combination of the columns $\mathbf{a}_j$. It is a slight change of notation to highlight that $\mathbf{x}$ acts of $\mathbf{A}$ to produce $\mathbf{b}$. For example, we have

$$\begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1\times 3 + 0 \times 2 + 2 \times 1 \\ (-1) \times 3 + 3 \times 2 + 1 \times 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}. \tag{3.22}$$

Note that once we have defined matrix-vector multiplication properly, the fact that any linear function can be written in the form (3.20) (for some particular matrix $\mathbf{A}$) is a nice generalization of the fact that any single linear function of a single variable has the form $f(x) = ax$ for some single number a. That simple case is just the case $m = n = 1$ in which the "matrix" a is 1×1.

Next we define the addition between two matrices and the scalar multiplication.

Definition 85. For any matrices $\mathbf{A} = (a_{ij})_{m\times n}$ and $\mathbf{B} = (b_{ij})_{m\times n}$ in $\mathbb{R}^{m\times n}$, the addition $\mathbf{A}+\mathbf{B}$ is a matrix in $\mathbb{R}^{m\times n}$ whose entries are the sum of the entries of $\mathbf{A}$ and $\mathbf{B}$, *i.e.* $\mathbf{A} + \mathbf{B} = (a_{ij} + b_{ij})_{m\times n}$.

Definition 86. For any matrix $\mathbf{A} = (a_{ij})_{m\times n}$ in $\mathbb{R}^{m\times n}$ and any scalar α in $\mathbb{R}$, the scalar multiplication $\alpha\mathbf{A}$ is a matrix in $\mathbb{R}^{m\times n}$ whose entries are the entries of $\mathbf{A}$ multiplied by α, *i.e.* $\alpha\mathbf{A} = (\alpha a_{ij})_{m\times n}$.

An alternative definition of the addition of two matrices uses linear functions. Suppose we have two linear functions $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ and $\mathbf{g}(\mathbf{x}) = \mathbf{B}\mathbf{x}$ that are defined by two different matrices $\mathbf{A}$ and $\mathbf{B}$ (both of which are $m \times n$, so both $\mathbf{f}$ and $\mathbf{g}$ map $\mathbb{R}^n$ to $\mathbb{R}^m$). Now define a new function $\mathbf{h}(\mathbf{x})$ by

$$\mathbf{h}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}).$$

This means that, for any vector $\mathbf{x} \in \mathbb{R}^n$, to compute the value of $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^m$, we first compute the two vectors $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$. Then we add them together, using the "vector addition" rule from $\mathbb{R}^m$. So the ith component of $\mathbf{h}(\mathbf{x})$ is just the ith component of $\mathbf{f}(\mathbf{x})$ added to the ith component of $\mathbf{g}(\mathbf{x})$.

Since $\mathbf{f}$ and $\mathbf{g}$ are both linear functions, it turns out that the function $\mathbf{h}$ is also a linear function. To see this, note that the ith component of $\mathbf{h}(\mathbf{x})$ is

$$\begin{aligned} h_i(\mathbf{x}) &= f_i(\mathbf{x}) + g_i(\mathbf{x}) \\ &= \sum_{j=1}^{n} a_{ij}x_j + \sum_{j=1}^{n} b_{ij}x_j \\ &= \sum_{j=1}^{n} (a_{ij} + b_{ij})x_j. \end{aligned} \tag{3.23}$$

But this means that $\mathbf{h}(\mathbf{x})$ is defined by the matrix-vector multiplication

$$\mathbf{h}(\mathbf{x}) = \mathbf{C}\mathbf{x}$$

where $\mathbf{C}$ is the $m \times n$ matrix with components $c_{ij} = a_{ij} + b_{ij}$. In other words,

$$\mathbf{C} = \mathbf{A} + \mathbf{B}$$

where we define the sum of two matrices of the same shape in the obvious way, by adding the corresponding elements of the two matrices. So $\mathbf{h}$ is a linear function and the matrix that defines it is simply the sum of the matrices $\mathbf{A}$ and $\mathbf{B}$ defining the functions $\mathbf{f}$ and $\mathbf{g}$.

Proposition 87. *$\mathbb{R}^{m\times n}$ is a real linear space.*

Remark. The set of all matrices with complex entries, m rows, and n columns is denoted $\mathbb{C}^{m\times n}$, which is a complex linear space.

Next, we give a proof that $\mathbb{R}^{m\times n}$ is a real linear space.

1. For any matrix $\mathbf{A}$ and any matrix $\mathbf{B}$ in $\mathbb{R}^{m\times n}$, we have

   $$\mathbf{A} = (a_{ij})_{m\times n} \quad \mathbf{B} = (b_{ij})_{m\times n} \;\Rightarrow\; \mathbf{A} + \mathbf{B} = (a_{ij} + b_{ij})_{m\times n}.$$

   So the matrix $\mathbf{A} + \mathbf{B}$ is also a matrix with m rows and n columns and it belongs to $\mathbb{R}^{m\times n}$.

2. For any matrix $\mathbf{A}$ and any matrix $\mathbf{B}$ in $\mathbb{R}^{m\times n}$, the vectors $\mathbf{A}+\mathbf{B}$ and $\mathbf{B}+\mathbf{A}$ are equal. Indeed, we have

$$\mathbf{A}+\mathbf{B}=(a_{ij}+b_{ij})_{m\times n}\quad \mathbf{B}+\mathbf{A}=(b_{ij}+a_{ij})_{m\times n}$$

and the mn components are equal because the addition of scalar numbers is commutative.

3. The addition of matrices is associative. Indeed, we have

$$(\mathbf{A}+\mathbf{B})+\mathbf{C}=((a_{ij}+b_{ij})+c_{ij})_{m\times n}\quad \mathbf{A}+(\mathbf{B}+\mathbf{C})=(a_{ij}+(b_{ij}+c_{ij}))_{m\times n}\,.$$

The mn components are equal because the addition of scalar numbers is associative.

4. The zero matrix in $\mathbb{R}^{m\times n}$ is the matrix with all its mn components equal to 0,

$$\mathbf{0}=\begin{bmatrix}0&\cdots&0\\ \vdots&&\vdots\\ 0&\cdots&0\end{bmatrix}.$$

5. For any matrix $\mathbf{A}$ in $\mathbb{R}^{m\times n}$, the matrix $\mathbf{B}$, defined by

$$\mathbf{A}=(a_{ij})_{m\times n}\quad \mathbf{B}=(-a_{ij})_{m\times n}\,,$$

is such that $\mathbf{A}+\mathbf{B}=\mathbf{0}$. The matrix $\mathbf{B}$ is the additive inverse of $\mathbf{A}$.

6. For any matrix $\mathbf{A}$ in $\mathbb{R}^{m\times n}$ and any scalar α in $\mathbb{R}$, we have

$$\mathbf{A}=(a_{ij})_{m\times n}\ \Rightarrow\ \alpha\mathbf{A}=(\alpha a_{ij})_{m\times n}\,.$$

So the matrix $\alpha\mathbf{A}$ is also a matrix with m rows and n columns and it belongs to $\mathbb{R}^{m\times n}$.

7. For any matrix $\mathbf{A}$ in $\mathbb{R}^{m\times n}$ and any scalar α and β in $\mathbb{R}$, we have

$$\begin{aligned}(\alpha+\beta)\mathbf{A}=((\alpha+\beta)\times a_{ij})_{m\times n}&=(\alpha a_{ij}+\beta a_{ij})_{m\times n}\\ &=(\alpha a_{ij})_{m\times n}+(\beta a_{ij})_{m\times n}=\alpha\mathbf{A}+\beta\mathbf{A}.\end{aligned}$$

8. For any matrices $\mathbf{A}$ and $\mathbf{B}$ in $\mathbb{R}^{m\times n}$ and any scalar α in $\mathbb{R}$, we have

$$\begin{aligned}\alpha(\mathbf{A}+\mathbf{B})=\alpha\,(a_{ij}+b_{ij})_{m\times n}&=(\alpha a_{ij}+\alpha b_{ij})_{m\times n}\\ &=(\alpha a_{ij})_{m\times n}+(\alpha b_{ij})_{m\times n}=\alpha\mathbf{A}+\alpha\mathbf{B}.\end{aligned}$$

9. For any matrix $\mathbf{A}$ in $\mathbb{R}^{m\times n}$ and any scalar α and β in $\mathbb{R}$, we have

$$(\alpha\beta)\mathbf{A}=((\alpha\beta)\times a_{ij})_{m\times n}=(\alpha\beta a_{ij})_{m\times n}=\alpha\,(\beta a_{ij})_{m\times n}=\alpha\,(\beta\mathbf{A})\,.$$

10. For any matrix $\mathbf{A}$ in $\mathbb{R}^{m\times n}$, we have

$$1\cdot\mathbf{A} = 1\cdot(a_{ij})_{m\times n} = (1\times a_{ij})_{m\times n} = (a_{ij})_{m\times n}\,.$$

So $\mathbb{R}^{m\times n}$ is a real linear space.

Remark. Note that $\mathbb{R}^m$ is the set of column vectors with m rows and 1 column. $\mathbb{R}^{m\times 1}$ denotes the set of matrices with m rows and 1 column.

Proposition 88. *The matrix-vector multiplication satisfies*

$$\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{x} = (\mathbf{A}+\mathbf{B})\mathbf{x} \tag{3.24}$$

and

$$\mathbf{A}\mathbf{x}+\mathbf{A}\mathbf{y} = \mathbf{A}(\mathbf{x}+\mathbf{y}). \tag{3.25}$$

The proofs simply use the associativity of real numbers to rewrite $a_{ij}x_j + b_{ij}x_j$ as $(a_{ij}+b_{ij})x_j$. Indeed we have

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \sum_{j=1}^n a_{1j}x_j \\ \vdots \\ \sum_{j=1}^n a_{mj}x_j \end{bmatrix} \quad \mathbf{B}\mathbf{x} = \begin{bmatrix} \sum_{j=1}^n b_{1j}x_j \\ \vdots \\ \sum_{j=1}^n b_{mj}x_j \end{bmatrix}$$

$$\Rightarrow\ \mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{x} = \begin{bmatrix} \sum_{j=1}^n a_{1j}x_j + \sum_{j=1}^n b_{1j}x_j \\ \vdots \\ \sum_{j=1}^n a_{mj}x_j + \sum_{j=1}^n b_{mj}x_j \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n (a_{1j}+b_{1j})x_j \\ \vdots \\ \sum_{j=1}^n (a_{mj}+b_{mj})x_j \end{bmatrix}$$

So we have proved something nontrivial about matrix-vector algebra using the rules of standard algebra of real numbers. It is also true for the other property of matrix-vector multiplication.

3.5 Linear differential operators

Recall that algebra and the idea of linearity applies to to situations other than vectors and matrices. Many of the fundamental concepts we learn about linear algebra for vectors and matrices carry over directly to functions and differential operators. Seeing these ideas in a different context now may help you to understand what "linearity" means more generally and why it is important.

Let us introduce the concept of a operator, which is just a function that takes a function as input and produces some other function as output. This gets a bit confusing so we call it an "operator" instead of a function. A differential operator computes the output function by combining various derivatives of the input function.

The simplest differential operator is just $D = \frac{d}{dx}$. For example, if $f(x) = x^3$ then $D(f)$ is the function $f'(x) = 3x^2$. The operator D is a linear operator, it satisfies the same linearity properties as in Definition 82, where $U = C^1(\mathbb{R},\mathbb{R})$ and $V = C^0(\mathbb{R},\mathbb{R})$.

The operator D is a linear operator because it is true that

$$D(u+v) = D(u)+D(v), \qquad \text{since } \frac{d}{dx}(u(x)+v(x)) = \frac{d}{dx}u(x)+\frac{d}{dx}v(x), \tag{3.26}$$

and

$$D(\alpha u) = \alpha D(u), \qquad \text{since } \frac{d}{dx}(\alpha u(x)) = \alpha\frac{d}{dx}u(x). \tag{3.27}$$

These linearity properties extend of course to arbitrary linear combinations. You are used to using this when finding the derivatives of complicated functions by splitting them up, *e.g.*

$$\frac{d}{dx}\left(5x^4 + 2\cos(6x)\right) = 5\frac{d}{dx}(x^4) + 2\frac{d}{dx}\cos(6x) = 20x^3 - 12\sin(6x).$$

Linearity was used in the first step.

The second derivative operator $\frac{d^2}{dx^2}$ and all higher order derivative operators are also linear operators. We obtain a general linear differential operator by taking a linear combination of these differential operators (and also the zero'th order derivative operator, which is just the identity operator that maps any function to itself). For example, the operator

$$L = 8\frac{d^2}{dx^2} - 4\frac{d}{dx} + 6I \tag{3.28}$$

is a linear differential operator, where I denotes the identity operator. Applying L to a function $u(x)$ results in the function

$$(Lu)(x) = 8u''(x) - 4u'(x) + 6u(x). \tag{3.29}$$

Now consider the differential equation

$$8u''(x) - 4u'(x) + 6u(x) = x^3 \tag{3.30}$$

for $0 \le x \le 1$ with $u(0) = 1$ and $u(1) = 3$. The problem is to find a funtion $u(x)$ that satisfies the equation (3.30) everywhere in the interval and also satisfies the two boundary conditions. This is a linear differential equation since it has the form $(Lu)(x) = g(x)$ where L is the linear operator of (3.29) and $g(x)$ is a given function $g(x) = x^3$.

3.6 Useful commands in MATLAB

Here are a few commands in MATLAB useful for this section.

- `A = [1, 2;3, 4;5, 6];` defines the matrix $\mathbf{A} \in \mathbb{R}^{3\times 2}$. Note that commas separate the column entries on a same row and that semi-columns finish every row.
- `A = zeros(m,n)` defines the matrix in $\mathbb{R}^{m\times n}$ with entries equal to 0.

- `A = ones(m,n)` defines the matrix in $\mathbb{R}^{m \times n}$ with entries equal to 1.
- `size(A)` returns the dimensions of `A` in a row vector `[m n]`.
- `A = rand(m,n)` makes a matrix of size $m \times n$ with random values, uniformly distributed between 0 and 1.
- `A = randn(m,n)` makes a matrix of size $m \times n$ with random values, normally distributed (with mean 0 and variance 1).
- `A*x` computes the matrix-vector product.
- `tr(A)` computes the trace of matrix $\mathbf{A}$.

3.7 Exercises

Exercise 89. Consider the following matrices

$$\mathbf{A} = \begin{bmatrix} 10 & 2 \\ -1 & 1 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 4 & 3 & -2 \\ 5 & -6 & 1 \end{bmatrix}, \ \mathbf{D} = \begin{bmatrix} 1 & -1 \\ 2 & \sqrt{2} \\ 3 & \sqrt{3} \end{bmatrix}$$

and vectors

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \mathbf{z} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

1. Calculate, by hand, the following quantities: $\mathbf{A} + \mathbf{B}$, $\mathbf{Ax}$, $\mathbf{By}$, $3\mathbf{Dx} + \mathbf{z}$, $\mathbf{y} + 2\mathbf{Cz}$.

Exercise 90. The following linear functions can be written as $f(\mathbf{x}) = \mathbf{Ax}$ for some matrix $\mathbf{A}$ (not necessarily square). In each case, determine $\mathbf{A}$.

1. $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} 3x_1 - 2x_2 \\ 4x_2 - x_3 \end{bmatrix}, \quad \forall \mathbf{x} \in \mathbb{R}^3.$
2. $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_3 \\ x_4 \\ x_2 \\ x_1 \end{bmatrix}, \quad \forall \mathbf{x} \in \mathbb{R}^4.$ (The matrix is called a permutation matrix)
3. $f(\mathbf{x}) = 2x_1 - 4x_3 + 5x_4, \quad \forall \mathbf{x} \in \mathbb{R}^4.$
4. $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} 2x_1 \\ 0 \\ x_1 \\ 3x_1 \\ x_1 \end{bmatrix}, \quad \forall \mathbf{x} \in \mathbb{R}^1.$
5. $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_1 \cos\theta + x_3 \sin\theta \\ x_2 \\ -x_1 \sin\theta + x_3 \cos\theta \end{bmatrix}, \quad \forall \mathbf{x} \in \mathbb{R}^3.$ (The matrix is called a rotation matrix)

Exercise 91. Consider the function $f(x) = 3x^4 + x\sin(2\pi x)$. Is the function f linear? Motivate your answer. With MATLAB, plot the function f over the interval $[-1, 1]$. On the same plot, plot the function $g(x) = f(0.8) + f'(0.8)(x - 0.8)$ with a dashed line. Is the function g linear?

Exercise 92. For each linear space below, determine a basis and the dimension of the space:

1. The space of all 2×2 matrices of the form

$$\mathbf{A} = \begin{bmatrix} a & b \\ b & a \end{bmatrix} \text{ for some } a, b \in \mathbb{R}$$

2. The space of all 4×4 diagonal matrices (matrices $\mathbf{A} \in \mathbb{R}^{4\times 4}$ for which $a_{ij} = 0$ if $i \neq j$)

3. The space of all 3×3 symmetric matrices (matrices $\mathbf{A} \in \mathbb{R}^{3\times 3}$ for which $a_{ij} = a_{ji}$). For example, the following matrices are in this space:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 3 & 4 & 5 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} -1 & 2 & 0 \\ 2 & 1 & 4 \\ 0 & 4 & 0 \end{bmatrix}$$

4. The space of all 3×3 anti-symmetric matrices (matrices $\mathbf{A} \in \mathbb{R}^{3\times 3}$ for which $a_{ij} = -a_{ji}$). For example, the following matrix is anti-symmetric:

$$\mathbf{A} = \begin{bmatrix} 0 & -2 & 3 \\ 2 & 0 & -4 \\ -3 & 4 & 0 \end{bmatrix}$$

5. The space of all 6×6 tridiagonal matrices. A tridiagonal matrix is of the form

$$\mathbf{A} = \begin{bmatrix} a & b & 0 & 0 & 0 & 0 \\ c & a & b & 0 & 0 & 0 \\ 0 & c & a & b & 0 & 0 \\ 0 & 0 & c & a & b & 0 \\ 0 & 0 & 0 & c & a & b \\ 0 & 0 & 0 & 0 & c & a \end{bmatrix} \text{ for some } a, b, c \in \mathbb{R}$$

Exercise 93. Consider the matrices $\mathbf{A} = (a_{ij})_{m\times n}$ and $\mathbf{B} = (b_{ij})_{m\times n}$. Let $\alpha, \beta \in \mathbb{R}$. What are the entries of the matrix $\alpha\mathbf{A} + \beta\mathbf{B}$?

Exercise 94. Consider the square matrix $\mathbf{A} = (a_{ij})_{n\times n}$, *i.e.* $m = n$. The function,

$$\operatorname{tr}(\mathbf{A}) = a_{11} + a_{22} + \cdots + a_{nn} = \sum_{i=1}^{n} a_{ii}, \tag{3.31}$$

evaluates the trace of a square matrix. Show that the trace function is a linear function on $\mathbb{R}^{n\times n}$.

Exercise 95. Check whether the following functions are linear or not. When the function is linear, determine the matrix $\mathbf{A}$ such that $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$.

1. $\forall x \in \mathbb{R},\ f(x) = 0$
2. $\forall x \in \mathbb{R},\ f(x) = |x|$
3. $\forall x \in \mathbb{R},\ f(x) = |x| + x$
4. $\forall x \in \mathbb{R},\ f(x) = (x-1)^2 + (x+1)^2$
5. $\forall x \in \mathbb{R},\ f(x) = (x+1)^2 - (x-1)^2$

6. $\forall x \in \mathbb{R},\ f(x) = e^x$
7. $\forall x \in \mathbb{R},\ f(x) = \log(x)$
8. $\forall x \in \mathbb{R},\ f(x) = \log(e^x)$
9. $f(x) = \begin{cases} 2x & \text{if } x \leq 1 \\ |2x| & \text{if } x > 1 \end{cases}$
10. $\forall \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3,\ f(\mathbf{x}) = x_1 + 2 - x_3$
11. $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_1 & x_1 + x_2 \\ x_1 + x_2 & x_1 \end{bmatrix}, \quad \forall \mathbf{x} \in \mathbb{R}^2.$

Exercise 96. Consider the operator $\mathcal{O} : C^0(\mathbb{R}, \mathbb{R}) \to C^1(\mathbb{R}, \mathbb{R})$ defined by

$$\mathcal{O}(f)(x) = \int_0^x f(t)dt \quad \forall x \in \mathbb{R}.$$

Check whether the operator $\mathcal{O}$ is linear.

Exercise 97. Consider the operator $\mathcal{O} : C^0(\mathbb{R}, \mathbb{R}) \to C^0(\mathbb{R}, \mathbb{R})$ defined by

$$\mathcal{O}(f)(x) = xf(x) \quad \forall x \in \mathbb{R}.$$

Check whether the operator $\mathcal{O}$ is linear.

Exercise 98. Consider the operator $\mathcal{O} : C^1(\mathbb{R}, \mathbb{R}) \to C^0(\mathbb{R}, \mathbb{R})$ defined by

$$\mathcal{O}(f)(x) = 2f'(x)f(x) \quad \forall x \in \mathbb{R}.$$

Check whether the operator $\mathcal{O}$ is linear.

Exercise 99. Determine whether the following operators D are linear. If not, find a counterexample. In each case we assume $u(x)$ is a differentiable function and $D(u)$ is another function given below:

- $D(u)(x) = u(x) + 2u''(x)$
- $D(u)(x) = u(x) + x$
- $D(u)(x) = 2u'(x) + \int_0^x u(t)dt$

4 Matrices

4.1 Space of $m \times n$ matrices

The set of $m \times n$ real matrices is denoted $\mathbb{R}^{m\times n}$. The previous section proved that $\mathbb{R}^{m\times n}$ is a real linear space. Its dimension is equal to mn. A simple basis for $\mathbb{R}^{m\times n}$ consists of the mn matrices with only one non-zero entry.

For example, consider $\mathbb{R}^{3\times 2}$. Any matrix with 3 columns and 2 rows is written as follows

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}, \quad \text{where} \quad a_{11},\, a_{12},\, a_{21},\, a_{22},\, a_{31},\, a_{32} \in \mathbb{R}.$$

The matrix $\mathbf{A}$ depends on 6 parameters: a_{11}, a_{12}, a_{21}, a_{22}, a_{31}, and a_{32}. We write a linear combination of matrices where each parameter is multiplying a matrix

$$\mathbf{A} = a_{11}\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + a_{12}\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + a_{21}\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} + a_{22}\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} + a_{31}\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} + a_{32}\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

This last formula illustrates that the 6 matrices span $\mathbb{R}^{3\times 2}$,

$$\mathbb{R}^{3\times 2} = \operatorname{span}\left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \right).$$

These 6 matrices are also linearly independent. Indeed, we have

$$\alpha\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + \beta\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + \gamma\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} + \delta\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} + \epsilon\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} + \zeta\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \\ \epsilon & \zeta \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

By identifying each entry to zero, we obtain the following 6 equations

$$\begin{cases} \alpha &= 0 \quad \text{(from entry (1,1))} \\ \beta &= 0 \quad \text{(from entry (1,2))} \\ \gamma &= 0 \quad \text{(from entry (2,1))} \\ \delta &= 0 \quad \text{(from entry (2,2))} \\ \epsilon &= 0 \quad \text{(from entry (3,1))} \\ \zeta &= 0 \quad \text{(from entry (3,2))} \end{cases}$$

So these 6 matrices are linearly independent because the only linear combination resulting in the zero matrix is the trivial combination with

$$\alpha = \beta = \gamma = \delta = \epsilon = \zeta = 0.$$

Consequently, a basis for $\mathbb{R}^{3\times 2}$ is given by

$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \right). \tag{4.1}$$

The dimension of $\mathbb{R}^{3\times 2}$ is 6.

Remark. The set of $m \times n$ complex matrices is denoted $\mathbb{C}^{m\times n}$. It is a real linear space of dimension $2mn$.

Remark. The set of $m \times n$ complex matrices is denoted $\mathbb{C}^{m\times n}$. It is a complex linear space of dimension mn.

4.2 Matrix-matrix multiplication

If $\mathbf{A} \in \mathbb{R}^{m\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times n}$, then we can define the product $\mathbf{C} = \mathbf{AB}$, which will be a matrix in $\mathbb{R}^{m\times n}$. Note that this product is only defined if the number of columns in $\mathbf{A}$ is equal to the number of rows in $\mathbf{B}$.

The elements of the product matrix $\mathbf{C}$ are given by

$$c_{ij} = \sum_{k=1}^{r} a_{ik} b_{kj}. \tag{4.2}$$

For example, if

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix}, \tag{4.3}$$

then $\mathbf{C} \in \mathbb{R}^{3\times 4}$ and

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ c_{41} & c_{42} & c_{43} \end{bmatrix}, \tag{4.4}$$

where, for example,

$$\begin{aligned} c_{11} &= a_{11}b_{11} + a_{12}b_{21}, \\ c_{12} &= a_{11}b_{12} + a_{12}b_{22}, \\ &\vdots \\ c_{43} &= a_{41}b_{13} + a_{42}b_{23}. \end{aligned}$$

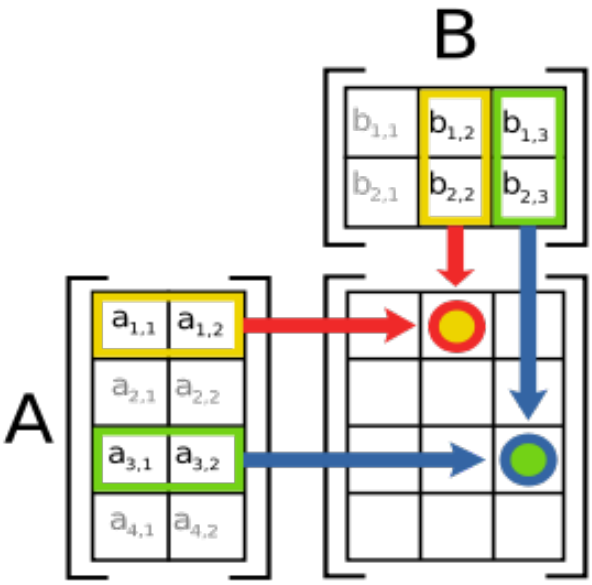


Figure 4.1: Illustration of Matrix-Matrix product (Diagram from Wikipedia).

$$\begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 \times 3 + 0 \times 2 + 2 \times 1 & 1 \times 1 + 0 \times 1 + 2 \times 0 \\ -1 \times 3 + 3 \times 2 + 1 \times 1 & -1 \times 1 + 3 \times 1 + 1 \times 0 \end{bmatrix} = \begin{bmatrix} 5 & 1 \\ 4 & 2 \end{bmatrix}$$

Figure 4.2: Example of Matrix-Matrix product (Diagram from Wikipedia).

The matrix-matrix product is illustrated in Figure 4.1. A numerical example is depicted in Figure 4.2. The diagram highlights the origins of each coefficient.

Written in terms of columns, the product is

$$\left[\begin{array}{c|c|c} & & \\ \mathbf{c}_1 & \cdots & \mathbf{c}_n \\ & & \end{array}\right] = \mathbf{A} \left[\begin{array}{c|c|c} & & \\ \mathbf{b}_1 & \cdots & \mathbf{b}_n \\ & & \end{array}\right] = \left[\begin{array}{c|c|c} & & \\ \mathbf{A}\mathbf{b}_1 & \cdots & \mathbf{A}\mathbf{b}_n \\ & & \end{array}\right]. \tag{4.5}$$

Remark. In the case where $n = 1$, $\mathbf{B}$ only has one column, the matrix-matrix multiplication $\mathbf{AB}$ agrees with matrix-vector multiplication.

Remark. Yet another way to view matrix-matrix multiplication is in terms of rows. The ith row of the product $\mathbf{C} = \mathbf{AB}$ is the ith row of $\mathbf{A}$ multiplied by the matrix $\mathbf{B}$.

Suppose $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ are both square and of the same size. Then the products $\mathbf{AB}$ and $\mathbf{BA}$ are both defined. Each product is again an $n \times n$ matrix. Note, however, that in general $\mathbf{AB} \neq \mathbf{BA}$. Matrix multiplication is not commutative in general!

Example 100. Let

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 4 & 5 \\ 1 & 0 \end{bmatrix}. \tag{4.6}$$

Then

$$\mathbf{AB} = \begin{bmatrix} 6 & 5 \\ 3 & 0 \end{bmatrix}, \qquad \mathbf{BA} = \begin{bmatrix} 4 & 23 \\ 1 & 2 \end{bmatrix}. \tag{4.7}$$

Definition 101. Consider $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$. When the products $\mathbf{AB}$ and $\mathbf{BA}$ are equal, we say that $\mathbf{A}$ and $\mathbf{B}$ commute.

Example 102. A particular example of matrix-matrix product is the outer product. Consider the product of an m-dimensional column vector, $\mathbf{u} \in \mathbb{R}^{m \times 1}$, with an n-dimensional row vector, $\mathbf{v} \in \mathbb{R}^{1 \times n}$. The outer product is an $m \times n$ matrix that can be written

$$\left[\begin{array}{c} \\ \mathbf{u} \\ \\ \end{array}\right] \left[\begin{array}{ccc} v_1 & \cdots & v_n \end{array}\right] = \left[\begin{array}{c|c|c} v_1\mathbf{u} & \cdots & v_n\mathbf{u} \end{array}\right] = \begin{bmatrix} u_1v_1 & \cdots & u_1v_n \\ \vdots & & \vdots \\ u_nv_1 & \cdots & u_nv_n \end{bmatrix} \tag{4.8}$$

The columns are all multiple of the same vector $\mathbf{u}$, and similarly, the rows are all multiple of the same vector $\mathbf{v}$.

4.3 Range and rank of a matrix

Definition 103. Let $\mathbf{a}^{(1)}, \cdots, \mathbf{a}^{(n)} \in \mathbb{R}^m$ denote the columns of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. The column space of $\mathbf{A}$, also called the range of $\mathbf{A}$, is the subspace of $\mathbb{R}^m$ spanned by the columns of $\mathbf{A}$. It is denoted by

$$\mathcal{R}(\mathbf{A}) = \text{span}(\mathbf{a}^{(1)}, \cdots, \mathbf{a}^{(n)}). \tag{4.9}$$

Definition 104. The dimension of the range of $\mathbf{A}$ is called the rank of $\mathbf{A}$:

$$\text{rank}(\mathbf{A}) = \dim(\mathcal{R}(\mathbf{A})). \tag{4.10}$$

Note the following:

- $\text{rank}(\mathbf{A}) \leq m$, since $\mathcal{R}(\mathbf{A})$ is a subspace of $\mathbb{R}^m$;
- $\text{rank}(\mathbf{A}) \leq n$, since $\mathcal{R}(\mathbf{A})$ is spanned by the n columns of $\mathbf{A}$. So a basis for $\mathcal{R}(\mathbf{A})$ has at most n vectors;
- $\text{rank}(\mathbf{A}) = n$ if and only if the columns of $\mathbf{A}$ are linearly independent. In this case, the columns form a basis for $\mathcal{R}(\mathbf{A})$;
- if $\text{rank}(\mathbf{A}) < n$, then the columns are linearly dependent and there exists a vector $\mathbf{z} \neq \bar{0}$ such that $\mathbf{Az} = \bar{0} = \mathbf{0}_{m \times 1}$.

Similarly, the row rank of a matrix is the dimension of of the space spanned by its rows. Row rank always equals column rank. So we refer to this number simply as the rank of a matrix. An $m \times n$ matrix of full rank is a matrix with the maximal possible rank ($\min(m, n)$). For example, a matrix of full rank with $m \geq n$ must have n linearly independent columns.

Remark. The rank is defined similarly for complex matrices in $\mathbb{C}^{m \times n}$.

Example 105. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 2 & 6 \\ -5 & 3 \\ 1 & 2 \end{bmatrix}.$$

We denote $\mathbf{a}_1$ and $\mathbf{a}_2$ the two columns of $\mathbf{A}$. The range of $\mathbf{A}$ is spanned by these two column vectors:

$$\mathcal{R}(\mathbf{A}) = \text{span}(\mathbf{a}_1, \mathbf{a}_2).$$

To find the dimension of $\mathcal{R}(\mathbf{A})$, we need to find a basis. The dimension will be the number of vectors in the basis. To get a basis, we beed to find linearly independent vectors that span the range of $\mathbf{A}$. Assume there exists a linear combination of $\mathbf{a}_1$ and $\mathbf{a}_2$ equal to the zero vector:

$$\alpha \begin{bmatrix} 2 \\ -5 \\ 1 \end{bmatrix} + \beta \begin{bmatrix} 6 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{cases} 2\alpha + 6\beta = 0 \\ -5\alpha + 3\beta = 0 \\ \alpha + 2\beta = 0 \end{cases}$$

The last equation gives that $\alpha = -2\beta$. Replacing this value into the first equation gives $6\beta - 4\beta = 0$, which implies that $\beta = 0$. Consequently, we have $\alpha = \beta = 0$ and the two column vectors are linearly independent. So $(\mathbf{a}_1, \mathbf{a}_2)$ is a basis for $\mathcal{R}(\mathbf{A})$. The rank of $\mathbf{A}$ is 2.

Example 106. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 2 & 4 \end{bmatrix}.$$

We denote $\mathbf{a}_1$ and $\mathbf{a}_2$ the two columns of $\mathbf{A}$. The range of $\mathbf{A}$ is spanned by these two column vectors:

$$\mathcal{R}(\mathbf{A}) = \text{span}(\mathbf{a}_1, \mathbf{a}_2).$$

To find the dimension of $\mathcal{R}(\mathbf{A})$, we need to find a basis. The dimension will be the number of vectors in the basis. To get a basis, we beed to find linearly independent vectors that span the range of $\mathbf{A}$. Assume there exists a linear combination of $\mathbf{a}_1$ and $\mathbf{a}_2$ equal to the zero vector:

$$\alpha \begin{bmatrix} 3 \\ 2 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{cases} 3\alpha = 0 \\ 2\alpha + 4\beta = 0 \end{cases}$$

The first equation gives that $\alpha = 0$. Replacing this value into the last equation gives $\beta = 0$. The two column vectors are linearly independent. So $(\mathbf{a}_1, \mathbf{a}_2)$ is a basis for $\mathcal{R}(\mathbf{A})$. The rank of $\mathbf{A}$ is 2.

Example 107. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & -1 \\ 2 & -8 & 2 \\ -1 & -2 & 1 \end{bmatrix}.$$

We denote $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ the three columns of $\mathbf{A}$. The range of $\mathbf{A}$ is spanned by these three column vectors:

$$\mathcal{R}(\mathbf{A}) = \text{span}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3).$$

To find the dimension of $\mathcal{R}(\mathbf{A})$, we need to find a basis. The dimension will be the number of vectors in the basis. To get a basis, we beed to find linearly independent vectors that span the range of $\mathbf{A}$. Assume there exists a linear combination of $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ equal to the zero vector:

$$\alpha \begin{bmatrix} 3 \\ 2 \\ -1 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ -8 \\ -2 \end{bmatrix} + \gamma \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{cases} 3\alpha - \gamma = 0 \\ 2\alpha - 8\beta + 2\gamma = 0 \\ -\alpha - 2\beta + \gamma = 0 \end{cases}$$

The first equation gives that $\gamma = 3\alpha$. Replacing this value into the second and third equation gives

$$\begin{cases} \gamma = 3\alpha \\ 2\alpha - 8\beta + 6\alpha = 0 \\ -\alpha - 2\beta + 3\alpha = 0 \end{cases} \quad \Rightarrow \quad \begin{cases} \gamma = 3\alpha \\ 8\alpha - 8\beta = 0 \\ 2\alpha - 2\beta = 0 \end{cases}$$

which implies that $\beta = \alpha$ but leaves the value of α arbitrary. For example, we can choose $\alpha = 1$, $\beta = 1$, and $\gamma = 3$. Then we have

$$1 \begin{bmatrix} 3 \\ 2 \\ -1 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ -8 \\ -2 \end{bmatrix} + 3 \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 3+0-3 \\ 2-8+6 \\ -1-2+3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

These three vectors are linearly dependent. So the rank of $\mathbf{A}$ is smaller than 3. To check whether the rank is 2, we need to find two vectors that are linearly independent. Three choices are possible. Here we choose $\mathbf{a}_1$ and $\mathbf{a}_2$. Assume there exists a linear combination of$\mathbf{a}_1$ and $\mathbf{a}_2$ equal to the zero vector:

$$\alpha \begin{bmatrix} 3 \\ 2 \\ -1 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ -8 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{cases} 3\alpha = 0 \\ 2\alpha - 8\beta = 0 \\ -\alpha - 2\beta = 0 \end{cases}$$

which implies that $\alpha = 0$. Then we get $\alpha = \beta = 0$ and the two column vectors are linearly independent. So $(\mathbf{a}_1, \mathbf{a}_2)$ is a basis for $\mathcal{R}(\mathbf{A})$. The rank of $\mathbf{A}$ is 2.

4.4 Null space of a matrix

Note that the set of all vectors $\mathbf{z}$ for which $\mathbf{Az} = \mathbf{0}$ is a linear space. This set is included in $\mathbb{R}^n$, which is a real linear space. If $\mathbf{Az} = \mathbf{0}$ and $\mathbf{Aw} = \mathbf{0}$, then

$$\mathbf{A}(\alpha\mathbf{z} + \beta\mathbf{w}) = \alpha\mathbf{Az} + \beta\mathbf{Aw} = \mathbf{0}.$$

So the vector $\alpha\mathbf{z} + \beta\mathbf{w}$ belongs to this set. It is a subspace of $\mathbb{R}^n$ and a real linear space. This space is called the null space of $\mathbf{A}$.

Definition 108. The null space of $\mathbf{A} \in \mathbb{R}^{m\times n}$ is the subspace of $\mathbb{R}^n$ defined by

$$\mathcal{N}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{R}^n; \mathbf{A}\mathbf{z} = \mathbf{0}\}. \tag{4.11}$$

Note the following:

- if $\text{rank}(\mathbf{A}) = n$, then $\mathcal{N}(\mathbf{A}) = \{\mathbf{0}\}$;
- if $\text{rank}(\mathbf{A}) < n$, then $\mathcal{N}(\mathbf{A})$ has dimension at least 1. In fact, we can show

$$\dim(\mathcal{N}(\mathbf{A})) = n - \text{rank}(\mathbf{A}). \tag{4.12}$$

This result is sometimes referred to as the *rank-nullity* theorem or the *dimension* theorem.

Example 109. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 2 & 6 \\ -5 & 3 \\ 1 & 2 \end{bmatrix}.$$

We have seen that the rank of this matrix is 2. The rank-nullity theorem gives

$$\dim\mathcal{N}(\mathbf{A}) = 2 - 2 = 0.$$

The matrix $\mathbf{A}$ is full rank. The dimension of its null space is 0. We have $\mathcal{N}(\mathbf{A}) = \{\mathbf{0}\}$. There is no basis for this null space.

Example 110. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 2 & 4 \end{bmatrix}.$$

We have seen that the rank of this matrix is 2. The rank-nullity theorem gives

$$\dim\mathcal{N}(\mathbf{A}) = 2 - 2 = 0.$$

The matrix $\mathbf{A}$ is full rank. The dimension of its null space is 0. We have $\mathcal{N}(\mathbf{A}) = \{\mathbf{0}\}$. There is no basis for this null space.

Example 111. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & -1 \\ 2 & -8 & 2 \\ -1 & -2 & 1 \end{bmatrix}.$$

We have seen that the rank of this matrix is 2. The rank-nullity theorem gives

$$\dim\mathcal{N}(\mathbf{A}) = 3 - 2 = 1.$$

The dimension of its null space is 1. We have $\mathcal{N}(\mathbf{A}) = \text{span}(\mathbf{z})$, where $\mathbf{z}$ is a nonzero vector. Assume $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$. We have

$$\mathbf{Az} = \mathbf{0} \quad \Rightarrow \quad \begin{bmatrix} 3 & 0 & -1 \\ 2 & -8 & 2 \\ -1 & -2 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 3z_1 - z_3 \\ 2z_1 - 8z_2 + 2z_3 \\ -z_1 - 2z_2 + z_3 \end{bmatrix}$$

The first equation gives that $z_3 = 3z_1$. Replacing this value into the second and third equation gives

$$\begin{cases} z_3 = 3z_1 \\ 2z_1 - 8z_2 + 6z_1 = 0 \\ -z_1 - 2z_2 + 3z_1 = 0 \end{cases} \quad \Rightarrow \quad \begin{cases} z_3 = 3z_1 \\ 8z_1 - 8z_2 = 0 \\ 2z_1 - 2z_2 = 0 \end{cases}$$

which implies that $z_2 = z_1$ but leaves the value of z_1 arbitrary. It is normal that z_1 is arbitrary because the dimension of the null space is 1. So we should expect 1 parameter. For example, we can choose $z_1 = 1$, $z_2 = 1$, and $z_3 = 3$. Then we have

$$\begin{bmatrix} 3 & 0 & -1 \\ 2 & -8 & 2 \\ -1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 3+0-3 \\ 2-8+6 \\ -1-2+3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

This vector is nonzero. It spans the null space. So a basis of $\mathcal{N}(\mathbf{A})$ is $\left(\begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} \right)$.

Example 112. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Assume $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$ is a nonzero vector belonging to the null space of $\mathbf{A}$. We have

$$\mathbf{Az} = \mathbf{0} \quad \Rightarrow \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} z_1 + z_2 + z_3 \\ z_1 + z_2 + z_3 \\ z_1 + z_2 + z_3 \end{bmatrix}$$

All three equations give that $z_3 = -z_1 - z_2$. So a general formula for the null space of $\mathbf{A}$ is

$$\mathcal{N}(\mathbf{A}) = \left\{ \begin{bmatrix} z_1 \\ z_2 \\ -z_1 - z_2 \end{bmatrix} ; \, \forall z_1, z_2 \in \mathbb{R} \right\}.$$

We look for a basis for $\mathcal{N}(\mathbf{A})$. We notice that this formula depends on two parameters z_1 and z_2. We have

$$\forall \mathbf{z} \in \mathcal{N}(\mathbf{A}),\ \mathbf{z} = z_1 \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} + z_2 \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}.$$

So we can write

$$\mathcal{N}(\mathbf{A}) = \text{span}\left(\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} \right).$$

Next we check whether these two vectors are linearly independent. Consider a linear combination of these two vectors equal to the zero vector

$$\alpha \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{cases} \alpha = 0 \\ \beta = 0 \\ -\alpha - \beta = 0 \end{cases}$$

which implies that $\alpha = \beta = 0$. So these two vectors are linearly independent. They form a basis for $\mathcal{N}(\mathbf{A})$. So the dimension of the null space is 2. Using the rank nullity theorem, we get that the rank of $\mathbf{A}$ is equal to 1.

4.5 Transpose/Adjoint of a matrix

Definition 113. The transpose of an $m \times n$ matrix $\mathbf{A}$, denoted $\mathbf{A}^T$, is the $n \times m$ matrix whose (i, j) entry is the (j, i) entry of $\mathbf{A}$.

For example,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \quad \Longrightarrow \quad \mathbf{A}^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \end{bmatrix}. \tag{4.13}$$

Definition 114. When $\mathbf{A} = \mathbf{A}^T$, the matrix $\mathbf{A}$ is called symmetric. When $\mathbf{A} = -\mathbf{A}^T$, the matrix $\mathbf{A}$ is called skew-symmetric.

Note that $\text{rank}(\mathbf{A}^T) = \text{rank}(\mathbf{A})$, since the column rank and row rank are equal. However, the dimensions for $\mathcal{N}(\mathbf{A})$ and $\mathcal{N}(\mathbf{A}^T)$ can differ. For example, the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & -1 \\ 2 & 4 & 6 \end{bmatrix}$$

has a null space of dimension 1 and its transpose a null space of dimension 0. The null spaces are

$$\mathcal{N}(\mathbf{A}) = \text{span}\left(\begin{bmatrix} 1 \\ -5 \\ 3 \end{bmatrix} \right) \quad \mathcal{N}(\mathbf{A}^T) = \{\bar{0}\} = \{\mathbf{0}_{2\times 1}\}.$$

Remark 115. A symmetric matrix must be square.

Note that we have:

- $\left(\mathbf{A}^T\right)^T = \mathbf{A}$;
- the identity matrix is a symmetric matrix;
- a diagonal matrix, whose non-zero entries are only on the diagonal, is symmetric.

If $\mathbf{A} \in \mathbb{R}^{m\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times n}$, then the product $\mathbf{C} = \mathbf{AB} \in \mathbb{R}^{m\times n}$ exists. We can take the transpose of this matrix and will get an $n \times m$ matrix $\mathbf{C}^T \in \mathbb{R}^{n\times m}$. Note that in this case $\mathbf{B}^T \in \mathbb{R}^{n\times r}$ and $\mathbf{A}^T \in \mathbb{R}^{r\times m}$. So the matrix product $\mathbf{B}^T\mathbf{A}^T$ is defined and is an $n \times m$ matrix. In fact, this is just equal to $\mathbf{C}^T \in \mathbb{R}^{n\times m}$. So, in general, it is true that

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T. \tag{4.14}$$

The transpose of the product is the product of the transposes, but with the order reversed!

If $\mathbf{A} \in \mathbb{R}^{m\times n}$ is a matrix, then the product $\mathbf{A}^T\mathbf{A}$ exists and it is a square matrix of dimension n. The matrix $\mathbf{A}^T\mathbf{A}$ is a symmetric matrix. Indeed, we have

$$\left(\mathbf{A}^T\mathbf{A}\right)^T = (\mathbf{A})^T\left(\mathbf{A}^T\right)^T = \mathbf{A}^T\mathbf{A}.$$

Definition 116. The adjoint of a complex $m \times n$ matrix $\mathbf{A}$, denoted $\mathbf{A}^*$, is the $n \times m$ matrix whose (i, j) entry is the conjugate (j, i) entry of $\mathbf{A}$,

$$(\mathbf{A}^*)_{ij} = \overline{a_{ij}}, \tag{4.15}$$

obtained by negating the imaginary part of a_{ij}.

Definition 117. When $\mathbf{A} = \mathbf{A}^*$, the matrix $\mathbf{A}$ is called hermitian. When $\mathbf{A} = -\mathbf{A}^*$, the matrix $\mathbf{A}$ is called skew-hermitian.

Note that an hermitian matrix must be square. Similarly to the transpose, the adjoint of the product of matrices is the product of the adjoint, with the order reversed.

Consider $\mathbf{x} \in \mathbb{R}^3$,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},$$

then we have

$$\mathbf{x}^T\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_1^2 + x_2^2 + x_3^2.$$

We have proved that, for $\mathbf{x} \in \mathbb{R}^3$,

$$\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T\mathbf{x}}. \tag{4.16}$$

This relation is also true in $\mathbb{R}^m$. For the inner product defined in (1.14), we similarly prove that

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^m, \quad \mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y}. \tag{4.17}$$

In $\mathbb{C}^m$, we replace the transpose operation ^{T} with the adjoint operator *.

4.6 Matrix inverse

The $n \times n$ matrix that has 1's on the diagonal and 0's everywhere else is called the identity matrix and denoted by $\mathbf{I}$.

It is called the identity matrix because for any $\mathbf{x} \in \mathbb{R}^n$,

$$\mathbf{I}\mathbf{x} = \mathbf{x}.$$

Also, if $\mathbf{B} \in \mathbb{R}^{n \times n}$,

$$\mathbf{I}\mathbf{B} = \left[\begin{array}{c|c|c} \mathbf{I}\mathbf{b}_1 & \cdots & \mathbf{I}\mathbf{b}_n \end{array}\right] = \left[\begin{array}{c|c|c} \mathbf{b}_1 & \cdots & \mathbf{b}_n \end{array}\right] = \mathbf{B}.$$

So multiplying a matrix by $\mathbf{I}$ leaves it unchanged.

More generally, if $\mathbf{B} \in \mathbb{R}^{m \times n}$ then multiplying $\mathbf{B}$ on the left by an $m \times m$ identity matrix or on the right by an $n \times n$ identity matrix leaves $\mathbf{B}$ unchanged.

Definition 118. A nonsingular or invertible matrix is a square matrix of full rank.

Note that a full rank $n \times n$ square matrix has columns that form a basis for the linear space $\mathbb{R}^n$ (or $\mathbb{C}^n$ for complex matrices). Therefore, any vector in $\mathbb{R}^n$ has a unique expression as a linear combination of the column vectors. In particular, every basis vector $\mathbf{e}_j$ of the standard unit basis (2.15) has a unique expansion in column vectors,

$$\mathbf{e}_j = \sum_{i=1}^{n} \mathbf{a}_i z_{ij}. \tag{4.18}$$

Let $\mathbf{z}_j$ denote the column vector with entries z_{ij}. Then we have

$$\mathbf{e}_j = \mathbf{A}\mathbf{z}_j$$

and combining all the vectors $\mathbf{z}_j$ in the matrix $\mathbf{Z}$, we get

$$\left[\begin{array}{c|c|c} \mathbf{e}_1 & \cdots & \mathbf{e}_n \end{array}\right] = \mathbf{I} = \mathbf{A}\mathbf{Z}.$$

The matrix $\mathbf{Z}$ is called the inverse matrix of $\mathbf{A}$ and written as $\mathbf{A}^{-1}$. It should not be confused with the additive inverse of $\mathbf{A}$ that is $-\mathbf{A}$. The matrix $\mathbf{A}^{-1}$ is the inverse for the multiplication and satisfies the relation

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}. \tag{4.19}$$

It is the matrix version of the familiar expression

$$aa^{-1} = 1$$

for any nonzero scalar $a \in \mathbb{R}$.

Theorem 119. *For $\mathbf{A} \in \mathbb{R}^{n\times n}$, the following conditions are equivalent:*

- $\mathbf{A}$ *has a unique inverse* $\mathbf{A}^{-1}$ *such that* $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$,
- $\operatorname{rank}(\mathbf{A}) = n$,
- $\mathcal{R}(\mathbf{A}) = \mathbb{R}^n$,
- $\mathcal{N}(\mathbf{A}) = \{\mathbf{0}\}$.

A similar result holds for complex matrices when replacing $\mathbb{R}$ with $\mathbb{C}$.

For any nonzero scalar $a \in \mathbb{R}$, the expression

$$aa^{-1} = a^{-1}a = 1$$

holds. The same is true for invertible matrices, *i.e.*

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}. \tag{4.20}$$

Starting from (4.19), we multiply on the right by $\mathbf{A}$ to get

$$\mathbf{A}\mathbf{A}^{-1}\mathbf{A} = \mathbf{A} \quad \text{and} \quad \mathbf{A}\left(\mathbf{A}^{-1}\mathbf{A}\mathbf{x} - \mathbf{x}\right) = \mathbf{0} \quad \forall \mathbf{x} \in \mathbb{R}^n.$$

Since the null space of $\mathbf{A}$ is trivial, we get

$$\mathbf{A}^{-1}\mathbf{A}\mathbf{x} = \mathbf{x} \quad \forall \mathbf{x} \in \mathbb{R}^n,$$

proving that $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$. The inverse matrix commutes with $\mathbf{A}$ and the product in either order is the identity matrix.

Corollary 120. *If $\mathbf{A}$ is invertible, then $\mathbf{A}^{-1}$ is invertible and we have $\left(\mathbf{A}^{-1}\right)^{-1} = \mathbf{A}$.*

Recall, from Section 4.5, that the transpose of a matrix product is the inverted product of the transposes. A similar formula holds for inverses in the square case. If $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ are both nonsingular, then so is their product $\mathbf{AB}$ and

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}. \tag{4.21}$$

Again the order is reversed. Indeed, we can write

$$\mathbf{B}^{-1}\mathbf{A}^{-1}\mathbf{A}\mathbf{B} = \mathbf{B}^{-1}\left(\mathbf{A}^{-1}\mathbf{A}\right)\mathbf{B} = \mathbf{B}^{-1}\mathbf{I}\mathbf{B} = \mathbf{B}^{-1}\mathbf{B} = \mathbf{I}$$

and use the uniqueness of the inverse matrix. If either $\mathbf{A}$ or $\mathbf{B}$ is singular, then the products $\mathbf{AB}$ and $\mathbf{BA}$ will also be singular and noninvertible. We emphasize that the inverse of a matrix is defined only for square matrices. So equation (4.21) holds only when $\mathbf{A}$ and $\mathbf{B}$ are both invertible and, in particular, square.

Note that, for any square invertible matrix $\mathbf{A}$, we have

$$\left(\mathbf{A}^T\right)^{-1} = \left(\mathbf{A}^{-1}\right)^T. \tag{4.22}$$

Proposition 121. *The inverse of a 2×2 matrix is equal to*

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{4.23}$$

if and only if $ad - bc \neq 0$.

The scalar $ad - bc$ is called the determinant of the matrix,

$$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc. \tag{4.24}$$

A 2×2 matrix is invertible if and only if its determinant is nonzero. The determinant can be defined for any square $n \times n$ matrix. For example, for 3×3 matrices, we have

$$\begin{aligned}
\det\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}\right) &= a \det\left(\begin{bmatrix} e & f \\ h & i \end{bmatrix}\right) - b \det\left(\begin{bmatrix} d & f \\ g & i \end{bmatrix}\right) + c \det\left(\begin{bmatrix} d & e \\ g & h \end{bmatrix}\right) \\
\det\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}\right) &= a(ei - hf) - b(di - gf) + c(dh - ge) \\
\det\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}\right) &= (aei + bfg + cdh) - (ahf + bdi + cge)
\end{aligned}$$

We can generalize that a $n \times n$ matrix is invertible if and only if its determinant is nonzero. However, the determinant rarely finds a useful role in numerical algorithms.

The determinant satisfies the following properties:

- $\det(\mathbf{I}) = 1$;
- $\det(\mathbf{AB}) = \det(\mathbf{A}) \det(\mathbf{B})$;
- $\det(\mathbf{A}^T) = \det(\mathbf{A})$;
- $\det(\mathbf{A}^{-1}) = 1/\det(\mathbf{A})$;
- $\det(\alpha\mathbf{A}) = \alpha^n \det(\mathbf{A})$, where $\mathbf{A}$ is a square matrix of dimension n.

Remark. When writing the product $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, we should not think of $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ as the result of applying $\mathbf{A}^{-1}$ to $\mathbf{b}$. Instead, we should understand $\mathbf{x}$ as the unique vector that satisfies the equation $\mathbf{Ax} = \mathbf{b}$. This means that $\mathbf{x}$ is the vector of coefficients of the unique linear combination of $\mathbf{b}$ in the basis of columns of $\mathbf{A}$. Multiplication by $\mathbf{A}^{-1}$ is a change of basis operation.

Remark. Numerically, we rarely work with inverse matrices. If we knew the inverse matrix, then we could solve any linear system $\mathbf{Ax} = \mathbf{b}$ simply by multiplying $\mathbf{b}$ by $\mathbf{A}^{-1}$. However, in practice, there are better ways to solve the system (*e.g.*, Gaussian elimination) that require less work than computing the inverse matrix and often give more accurate solutions (when the rounding errors of computer arithmetic are taken into account).

4.7 Orthogonal/Unitary matrices

Orthogonal and unitary matrices play an important role in numerical algorithms. We give their definitions here. Some of their properties will be described later in the notes.

Definition 122. A square $n \times n$ real matrix $\mathbf{O}$ is orthogonal if $\mathbf{O}^T = \mathbf{O}^{-1}$ or $\mathbf{O}^T\mathbf{O} = \mathbf{I}$.

Definition 123. A square $n \times n$ complex matrix $\mathbf{Q}$ is unitary if $\mathbf{Q}^* = \mathbf{Q}^{-1}$ or $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$.

When $n = 2$, the orthogonal matrices are

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} \tag{4.25}$$

where θ belongs to $\mathbb{R}$.

Let $\mathbf{O}$ denote an $n \times n$ real orthogonal matrix and $\mathbf{x}$ an $n \times 1$ real vector. The 2-norm of $\mathbf{Ox}$ is equal to

$$\|\mathbf{Ox}\|_2 = \sqrt{(\mathbf{Ox})^T\mathbf{Ox}} = \sqrt{\mathbf{x}^T\mathbf{O}^T\mathbf{Ox}} = \sqrt{\mathbf{x}^T\mathbf{x}} = \|\mathbf{x}\|_2. \tag{4.26}$$

The multiplication by an orthogonal matrix preserves the 2-norm of a vector. Orthogonal matrices preserve also the inner product (1.14). Note that they do not preserve the ∞-norm or the 1-norm. For example,

$$\left\|\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right\|_\infty = 1 \quad \text{and} \quad \left\|\begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right\|_\infty = \frac{\sqrt{2}}{2}.$$

In $\mathbb{C}^m$, the unitary matrices preserve the 2-norm of complex vectors and the inner product.

4.8 Useful commands in MATLAB

Here are a few commands in MATLAB useful for this section.

- `C = A*B` computes the matrix-matrix product.
- `rank(A)` computes the rank of a matrix.
- `null(A)` generates an orthonormal basis for the null space of $\mathbf{A}$.
- `A'` computes the adjoint matrix $\mathbf{A}^*$. When $\mathbf{A}$ has only real entries, the adjoint matrix $\mathbf{A}^*$ is equal to the transpose matrix $\mathbf{A}^T$. To compute the matrix (without the complex conjuguate), MATLAB uses `A.'`.
- `I = eye(n)` creates the identity matrix in $\mathbb{R}^{n\times n}$.
- `inv(A)` computes the inverse of matrix $\mathbf{A}$ when it exists.
- `det(A)` computes the determinant of a matrix.

4.9 Exercises

Exercise 124. Compute the rank of the following matrices

$$\begin{bmatrix} 2 & 6 \\ 1 & 3 \\ 2/3 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix}.$$

Exercise 125. Compute the range and rank of an outer product, i.e. $\mathbf{u}\mathbf{v}^T$ where the vectors $\mathbf{u}$ and $\mathbf{v}$ are nonzero.

Exercise 126. Compute the null space of the following matrices

$$\begin{bmatrix} 2 & 6 \\ 1 & 3 \\ 2/3 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix}.$$

Exercise 127. Compute the rank, the dimension of the null space, and a basis of the nullspace for the following matrices:

- $\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & -1 & 5 \\ -1 & 0 & -2 \end{bmatrix}$
- $\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$
- $\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
- $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ -1 & -1 \\ 0 & 1 \end{bmatrix}$

Exercise 128. Describe the null space of an outer product, i.e. $\mathbf{u}\mathbf{v}^T$ where the vectors $\mathbf{u}$ and $\mathbf{v}$ are nonzero.

Exercise 129. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & 0 & 1 & 2 \\ 0 & 2 & -2 & 0 & 1 \\ 0 & 0 & 0 & 4 & 1 \end{bmatrix}.$$

Determine a basis for the spaces $\mathcal{R}(\mathbf{A})$ and $\mathcal{N}(\mathbf{A})$.

Exercise 130. Consider the matrix $\mathbf{A} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -2 \\ 1 & -2 & 3 \end{bmatrix}$. What are the rank and the nullity of $\mathbf{A}$?

Exercise 131. Determine a basis for the spaces $\mathcal{R}(\mathbf{A})$ and $\mathcal{N}(\mathbf{A})$ where

$$\mathbf{A} = \begin{bmatrix} 1 & -3 & 2 & 0 \\ -1 & 5 & 1 & 1 \\ 2 & -8 & 1 & -1 \end{bmatrix}.$$

Exercise 132. Let $\mathbf{A}$ be a square real matrix. Show that $\mathbf{A}+\mathbf{A}^T$ is a symmetric matrix. Show that $\mathbf{A}-\mathbf{A}^T$ is a skew-symmetric matrix. Prove that any square matrix is the sum of a symmetric and a skew-symmetric matrix.

Exercise 133. Characterize the diagonal entries of a real skew-symmetric matrix.

Exercise 134. Characterize the diagonal entries of a complex hermitian matrix.

Exercise 135. Prove that the set of all 4×4 matrices of the form

$$\mathbf{A} = \begin{bmatrix} a & d & c & b \\ b & a & d & c \\ c & b & a & d \\ d & c & b & a \end{bmatrix} \text{ for some } a, b, c, d \in \mathbb{R}$$

is a real linear space. Give a basis and the dimension.

Exercise 136. Prove formula

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{4.27}$$

Exercise 137. Consider a real matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$. The matrix $\mathbf{A}$ is of full rank, *i.e.* $\text{rank}(\mathbf{A}) = n$.

- Find the null space of $\mathbf{A}$.
- Show that, for any vector $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = \|\mathbf{A}\mathbf{x}\|_2^2$.
- Show that the matrix $\mathbf{A}^T\mathbf{A}$ is invertible.
- We define the Moore-Penrose pseudoinverse as the matrix $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$.
  1. Simplify the expression $\mathbf{A}\mathbf{A}^\dagger\mathbf{A}$
  2. Simplify the expression $\mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger$
  3. Show the symmetry property $\left(\mathbf{A}\mathbf{A}^\dagger\right)^T = \mathbf{A}\mathbf{A}^\dagger$
  4. Show the other symmetry property $\left(\mathbf{A}^\dagger\mathbf{A}\right)^T = \mathbf{A}^\dagger\mathbf{A}$

5 Norms and Inner Products

In this section, we generalize the norms and the inner product defined on $\mathbb{R}^m$ (see Section 1.3) to linear spaces. In particular, we discuss norm and inner products for matrices.

5.1 Norms

Definition 138. Consider U a real linear space. A norm $\|\cdot\|$ is a map from $U \to \mathbb{R}^+$, satisfying the following conditions :

$$\forall u \in U, \|u\| \geq 0, \tag{5.1a}$$
$$\|u\| = 0 \text{ if and only if } u = \bar{0}, \tag{5.1b}$$
$$\forall \alpha \in \mathbb{R}, \|\alpha u\| = |\alpha| \|u\|, \tag{5.1c}$$
$$\forall u^{(1)}, u^{(2)} \in U, \|u^{(1)} + u^{(2)}\| \leq \|u^{(1)}\| + \|u^{(2)}\| \quad \text{(Triangle inequality)}. \tag{5.1d}$$

To define a norm for matrices, several approaches are possible. First, we look at the matrix just through its entries. In that case, the resulting matrix norms are very similar to vector norms. For example, the map

$$\mathbf{A} = (a_{ij})_{m\times n} \longmapsto \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^p \right)^{1/p}$$

is a norm on $\mathbb{R}^{m\times n}$. Note that it is similar to the vector p-norm on $\mathbb{R}^{mn}$.

Example 139. Consider the linear space of real matrices $U = \mathbb{R}^{m\times n}$ and the matrix $\mathbf{A} = (a_{ij})_{m\times n}$. The map,

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}, \tag{5.2}$$

is called the Frobenius norm. It is similar to the 2-norm of $\mathbb{R}^{mn}$. It is easy to show that this map is a norm according to Definition 138. The Frobenius norm can be used to bound the product of matrices. Let $\mathbf{C} = \mathbf{AB}$ with entries c_{ij} and let $\mathbf{a}_{row,i}$ be the ith row of $\mathbf{A}$ and $\mathbf{b}_j$ the jth column of $\mathbf{B}$. Then we have $c_{ij} = \mathbf{a}_{row,i}\mathbf{b}_j$. So, by the Cauchy-Schwarz inequality, we have

$$|c_{ij}| \leq \|\mathbf{a}_{row,i}\|_2 \|\mathbf{b}_j\|_2.$$

Squaring both sides and summing over i and j, we get

$$\begin{aligned}\|\mathbf{AB}\|_F^2 &= \sum_{i=1}^{m}\sum_{j=1}^{n}|c_{ij}|^2, \\ &\leq \sum_{i=1}^{m}\sum_{j=1}^{n}\|\mathbf{a}_{row,i}\|_2^2\|\mathbf{b}_j\|_2^2, \\ &\leq \left(\sum_{i=1}^{m}\|\mathbf{a}_{row,i}\|_2^2\right)\left(\sum_{j=1}^{n}\|\mathbf{b}_j\|_2^2\right) = \|\mathbf{A}\|_F^2\|\mathbf{B}\|_F^2.\end{aligned}$$

Another approach to define a matrix norm is to assess the effect of the matrix on the length of a vector. This approach is intimately related to a vector norm. The resulting norm is called the induced or associated matrix norm. Considering all the nonzero vectors $\mathbf{x}$, the induced matrix norm compares the length of the vectors $\mathbf{x}$ and $\mathbf{Ax}$ and is equal to the largest possible ratio between these lengths.

Definition 140. Consider the linear space of real matrices $\mathbb{R}^{m\times n}$ and the vector norms $\|\cdot\|_{(n)}$ on the domain space $\mathbb{R}^n \to \mathbb{R}^+$ and $\|\cdot\|_{(m)}$ on the range space $\mathbb{R}^m \to \mathbb{R}^+$. The induced matrix norm is defined as

$$\|\mathbf{A}\|_{(m,n)} = \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{Ax}\|_{(m)}}{\|\mathbf{x}\|_{(n)}} = \max_{\|\mathbf{x}\|_{(n)}=1} \|\mathbf{Ax}\|_{(m)}. \tag{5.3}$$

The induced matrix norm is the largest value of the ratios $\|\mathbf{Ax}\|_{(m)}/\|\mathbf{x}\|_{(n)}$. It is easy to verify that $\|\cdot\|_{(m,n)}$ satisfy the properties (5.1) defining a norm. Note that we used the subscripts (n) and (m) to avoid any confusion with the vector p-norms.

Sometimes, the ratio $\|\mathbf{Ax}\|_{(m)}/\|\mathbf{x}\|_{(n)}$ is called the *amplification factor* or *gain* in the direction $\mathbf{x}$. So the norm $\|\mathbf{A}\|_{(m,n)}$ is the maximum gain over all directions.

Example 141. Consider the matrix $\mathbf{A} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$. We have the following relations using, for example, the 1-norm of the vectors, *i.e.* the (1,1) induced matrix norm:

- $\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \|\begin{bmatrix} 1 \\ 0 \end{bmatrix}\|_1 = 1 \quad \|\mathbf{A}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\|_1 = 2.$ The ratio is here 2.
- $\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \Rightarrow \quad \|\begin{bmatrix} 0 \\ 1 \end{bmatrix}\|_1 = 1 \quad \|\mathbf{A}\begin{bmatrix} 0 \\ 1 \end{bmatrix}\|_1 = 1.$ The ratio is here 1.
- $\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \Rightarrow \quad \|\begin{bmatrix} 1 \\ 1 \end{bmatrix}\|_1 = 2 \quad \|\mathbf{A}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\|_1 = 3.$ The ratio is here $3/2$.

- $\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} \quad \Rightarrow \quad \|\begin{bmatrix} 2 \\ 1 \end{bmatrix}\|_1 = 3 \quad \|\mathbf{A}\begin{bmatrix} 2 \\ 1 \end{bmatrix}\|_1 = 5.$ The ratio is here $5/3$.

From these examples, we see that the ratio depends on the vector $\mathbf{x}$. It seems that 2 will be the largest one. It is, indeed, the case. Consider an arbitrary vector $\mathbf{x}$. Then we have

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \|\mathbf{x}\|_1 = |x_1| + |x_2|$$

and

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} 2x_1 \\ x_2 \end{bmatrix} \quad \|\mathbf{A}\mathbf{x}\|_1 = 2|x_1| + |x_2|.$$

So the ratio is equal to

$$\frac{\|\mathbf{A}\mathbf{x}\|_1}{\|\mathbf{x}\|_1} = \frac{2|x_1| + |x_2|}{|x_1| + |x_2|} \leq \frac{2|x_1| + 2|x_2|}{|x_1| + |x_2|} = 2.$$

This bound shows that the ratio can not be larger than 2. Since we have found one vector for which the ratio is exactly 2, the (1,1) induced matrix norm for $\mathbf{A}$ is equal to 2. This norm value indicates that the matrix $\mathbf{A}$ can multiply the length of a vector by 2, at most. It could be less than 2 for some vectors. However, it does not say anything about the direction of the vector.

Example 142. Let $\mathbf{A}$ be an $m \times n$ matrix. The $(1,1)$ induced norm is often denoted $\|\mathbf{A}\|_{(1,1)} = \|\mathbf{A}\|_1$. This notation is not confusing with the vector 1-norm because $\mathbf{A}$ is a matrix. For any vector $\mathbf{x} \in \mathbb{R}^n$, we have

$$\|\mathbf{A}\mathbf{x}\|_1 = \|\sum_{j=1}^{n} \mathbf{a}_j x_j\|_1 \leq \sum_{j=1}^{n} |x_j| \|\mathbf{a}_j\|_1 \leq \|\mathbf{x}\|_1 \max_{1 \leq j \leq n} \|\mathbf{a}_j\|_1,$$

where $\mathbf{a}_j$ is the jth colulumn vector of $\mathbf{A}$. Note that $\|\mathbf{a}_j\|_1$is the vector 1-norm because $\mathbf{a}_j$ is a vector. Therefore, the induced matrix 1-norm satisfies

$$\|\mathbf{A}\|_1 \leq \max_{1 \leq j \leq n} \|\mathbf{a}_j\|_1.$$

By choosing $\mathbf{x} = \mathbf{e}_j$, the unit standard basis vector of , where j is maximizing $\|\mathbf{a}_j\|_1$, we attain the bound. Thus the matrix 1-norm is

$$\|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \|\mathbf{a}_j\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |a_{ij}|, \tag{5.4}$$

which is equal to the maximum column sum. We can write

$$\|\mathbf{A}\mathbf{x}\|_1 \leq \|\mathbf{A}\|_1 \|\mathbf{x}\|_1.$$

Example 143. Let $\mathbf{A}$ be an $m \times n$ matrix. The (∞, ∞) induced norm is often denoted $\|\mathbf{A}\|_{(\infty,\infty)} = \|\mathbf{A}\|_\infty$. This notation is not confusing with the vector ∞-norm because $\mathbf{A}$ is a matrix. The matrix ∞-norm is

$$\|\mathbf{A}\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^{n} |a_{ij}|, \tag{5.5}$$

which is equal to the maximum row sum. We can write

$$\|\mathbf{A}\mathbf{x}\|_\infty \leq \|\mathbf{A}\|_\infty \|\mathbf{x}\|_\infty.$$

When comparing equations (5.4) and (5.5), we remark that, for any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we have

$$\|\mathbf{A}\|_\infty = \|\mathbf{A}^T\|_1. \tag{5.6}$$

A similar property holds for complex matrices when using the adjoint matrix $\mathbf{A}^*$.

Example 144. Let $\mathbf{A}$ be an $m \times n$ matrix. The $(2, 2)$ induced norm is denoted $\|\mathbf{A}\|_{(2,2)} = \|\mathbf{A}\|_2$. This notation is not confusing with the vector 2-norm because $\mathbf{A}$ is a matrix. The matrix 2-norm is defined by

$$\|\mathbf{A}\|_2 = \max_{\mathbf{x} \neq 0} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \sqrt{\max_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}}. \tag{5.7}$$

We can write

$$\|\mathbf{A}\mathbf{x}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|_2.$$

Later in the notes, we will introduce the spectral radius that describes precisely the 2-norm of a matrix. For the moment, just remember that the 2-norm of a matrix $\mathbf{A}$ is related to the matrix $\mathbf{A}^T\mathbf{A}$. As a consequence of (5.7), we remark that the 2-norm of a real matrix is invariant under multiplication by orthogonal matrices,

$$\forall \mathbf{A} \in \mathbb{R}^{m \times n}, \quad \|\mathbf{O}\mathbf{A}\|_2 = \|\mathbf{A}\|_2, \quad \forall \mathbf{O} \in \mathbb{R}^{m \times m} \text{ satisfying } \mathbf{O}^T\mathbf{O} = \mathbf{I}. \tag{5.8}$$

The 2-norm of a complex matrix is invariant under multiplication by unitary matrices.

Similarly to norms in $\mathbb{R}^m$, all the matrix norms are related to each other. Recall that we have

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{n} \|\mathbf{x}\|_\infty, \tag{5.9}$$

for any vector $\mathbf{x} \in \mathbb{R}^n$. We can write also, for $\mathbf{x} \neq \mathbf{0}$,

$$\frac{1}{\sqrt{n}\|\mathbf{x}\|_\infty} \leq \frac{1}{\|\mathbf{x}\|_2} \leq \frac{1}{\|\mathbf{x}\|_\infty}.$$

For any vector $\mathbf{y} \in \mathbb{R}^m$, we have

$$\|\mathbf{y}\|_\infty \leq \|\mathbf{y}\|_2 \leq \sqrt{m} \|\mathbf{y}\|_\infty.$$

For any matrix in $\mathbb{R}^{m\times n}$, we have

$$\frac{1}{\sqrt{n}}\|\mathbf{A}\|_\infty \leq \|\mathbf{A}\|_2 \leq \sqrt{m}\|\mathbf{A}\|_\infty. \tag{5.10}$$

Let us prove the right side of the bound. Consider any non-zero vector $\mathbf{x} \in \mathbb{R}^n$. We have

$$\frac{\|\mathbf{Ax}\|_2}{\|\mathbf{x}\|_2} \leq \sqrt{m}\frac{\|\mathbf{Ax}\|_\infty}{\|\mathbf{x}\|_2} \leq \sqrt{m}\frac{\|\mathbf{Ax}\|_\infty}{\|\mathbf{x}\|_\infty} \leq \sqrt{m}\|\mathbf{A}\|_\infty \tag{5.11}$$

where we used twice the equation (5.9) and where the last inequality is due to the supremum. So we have

$$\forall\, \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0},\ \frac{\|\mathbf{Ax}\|_2}{\|\mathbf{x}\|_2} \leq \sqrt{m}\|\mathbf{A}\|_\infty.$$

Note that the supremum is also the minimal upper bound. So we obtain

$$\|\mathbf{A}\|_2 \leq \sqrt{m}\|\mathbf{A}\|_\infty.$$

Next, we consider the left hand side. We have

$$\frac{\|\mathbf{Ax}\|_\infty}{\|\mathbf{x}\|_\infty} \leq \frac{\|\mathbf{Ax}\|_2}{\|\mathbf{x}\|_\infty} \leq \frac{\|\mathbf{Ax}\|_2}{\frac{1}{\sqrt{n}}\|\mathbf{x}\|_2} \leq \sqrt{n}\|\mathbf{A}\|_2 \tag{5.12}$$

where we used twice the equation (5.9) and where the last inequality is due to the supremum. So we have

$$\forall\, \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0},\ \frac{\|\mathbf{Ax}\|_\infty}{\|\mathbf{x}\|_\infty} \leq \sqrt{n}\|\mathbf{A}\|_2.$$

Using again the supremum as the minimal upper bound, we obtain

$$\frac{1}{\sqrt{n}}\|\mathbf{A}\|_\infty \leq \|\mathbf{A}\|_2.$$

The induced matrix norm of a matrix product can also be bounded. Let $\mathbf{A}$ be an $m \times r$ real matrix and $\mathbf{B}$ be an $r \times n$ real matrix. For any $\mathbf{x} \in \mathbb{R}^n$, we have

$$\|\mathbf{ABx}\|_m \leq \|\mathbf{A}\|_{(m,r)}\|\mathbf{Bx}\|_r \leq \|\mathbf{A}\|_{(m,r)}\|\mathbf{B}\|_{(r,n)}\|\mathbf{x}\|_n.$$

Therefore, the induced matrix norm satisfy

$$\|\mathbf{AB}\|_{(m,n)} \leq \|\mathbf{A}\|_{(m,r)}\|\mathbf{B}\|_{(r,n)}. \tag{5.13}$$

In particular, we have

$$\left\{\begin{array}{rcl} \|\mathbf{AB}\|_1 & \leq & \|\mathbf{A}\|_1\|\mathbf{B}\|_1 \\ \|\mathbf{AB}\|_2 & \leq & \|\mathbf{A}\|_2\|\mathbf{B}\|_2 \\ \|\mathbf{AB}\|_\infty & \leq & \|\mathbf{A}\|_\infty\|\mathbf{B}\|_\infty \end{array}\right. \tag{5.14}$$

Remark. When U is a complex linear space, a norm remains a map from $U \to \mathbb{R}^+$. The notions introduced in this Section can be extended to $\mathbb{C}^{m\times n}$.

5.2 Inner products

Definition 145. Consider U a real linear space. An inner product $< \cdot, \cdot >$ is a map $U \times U \to \mathbb{R}$ satisfying the following properties

$$\forall u \in U, < u, u > \geq 0, \tag{5.15a}$$
$$< u, u > = 0 \text{ if and only if } u = \bar{0}, \tag{5.15b}$$
$$\forall \alpha \in \mathbb{R}, \forall u,\ v \in U, < \alpha u, v > = \alpha < u, v >, \tag{5.15c}$$
$$\forall u,\ v \in U, < u, v > = < v, u >, \tag{5.15d}$$
$$\forall u,\ v,\ w \in U, < u + v, w > = < u, w > + < v, w > . \tag{5.15e}$$

An inner product is a symmetric positive-definite bilinear form.

Example 146. The Euclidean inner product of two column vectors in $\mathbb{R}^m$ (1.14) satisfy the properties (5.15).

The Cauchy-Schwarz inequality holds for any inner product. For any vectors u and v of U, we write

$$|< u, v >| \leq \sqrt{< u, u >}\sqrt{< v, v >}, \tag{5.16}$$

with equality if and only if v is proportional to u. The proof uses

$$< u - \lambda v, u - \lambda v > \geq 0$$

with

$$\lambda = \frac{< u, v >}{< v, v >}.$$

Proposition 147. *Given U a real linear space and an inner product $< \cdot, \cdot >$ on U. The map from U to $\mathbb{R}^+$, defined by*

$$u \longmapsto \sqrt{< u, u >}, \tag{5.17}$$

is a norm.

Example 148. Consider $U = \mathbb{R}^{m \times n}$. The map

$$(\mathbf{A}, \mathbf{B}) \longmapsto \operatorname{tr}(\mathbf{A}^T \mathbf{B}) \tag{5.18}$$

is an inner product. It satisfies also

$$\operatorname{tr}(\mathbf{A}^T \mathbf{A}) = \|\mathbf{A}\|_F^2. \tag{5.19}$$

From this last expression, we remark that the Frobenius norm is also invariant under multiplication by orthogonal matrices,

$$\forall \mathbf{A} \in \mathbb{R}^{m \times n}, \quad \|\mathbf{OA}\|_F = \|\mathbf{A}\|_F, \quad \forall \mathbf{O} \in \mathbb{R}^{m \times m} \text{ satisfying } \mathbf{O}^T \mathbf{O} = \mathbf{I}. \tag{5.20}$$

The associated inner product is also invariant under multiplication by an orthogonal matrix,

$$\operatorname{tr}\left((\mathbf{OA})^T \mathbf{OB}\right) = \operatorname{tr}(\mathbf{A}^T \mathbf{O}^T \mathbf{OB}) = \operatorname{tr}(\mathbf{A}^T \mathbf{B}).$$

The concept of orthogonality also extends to general real linear spaces.

Definition 149. Let U be a real linear space. We say that $u, v \in U$ are orthogonal for the inner product $< \cdot, \cdot >$ when $< u, v >= 0$, *i.e.* their inner product is 0.

We conclude by giving the definition of an inner product on a complex linear space. The Cauchy-Schwarz inequality and the orthogonality still holds in the complex case.

Definition 150. Consider U a complex linear space. An inner product $< \cdot, \cdot >$ is a map $U \times U \to \mathbb{C}$ satisfying the following properties

$$\forall u \in U, < u, u > \geq 0, \tag{5.21a}$$
$$< u, u >= 0 \text{ if and only if } u = \bar{0}, \tag{5.21b}$$
$$\forall \alpha \in \mathbb{R}, \forall u,\ v \in U, < \alpha u, v >= \alpha < u, v >, \tag{5.21c}$$
$$\forall u,\ v \in U, < u, v >= \overline{< v, u >}, \tag{5.21d}$$
$$\forall u,\ v,\ w \in U, < u + v, w >=< u, w > + < v, w > . \tag{5.21e}$$

An inner product is a symmetric positive-definite sesquilinear form.

5.3 Errors

In order to discuss the accuracy of a numerical solution or the relative merits of one numerical method versus another, it is necessary to choose a manner of measuring the error. It may seem obvious what is meant by the error. But there are often many different ways to measure the error that can sometimes give quite different impressions as to the accuracy of an approximate solution.

Consider a problem to which the true answer is a vector $\mathbf{x} \in \mathbb{R}^m$. Denote an approximation by $\hat{\mathbf{x}}$. Then the error in this approximation is

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}.$$

5.3.1 Absolute error

A natural measure of this error would be the norm of $\mathbf{e}$,

$$\|\mathbf{e}\| = \|\mathbf{x} - \hat{\mathbf{x}}\|.$$

This is called the *absolute* error in the approximation.

As an example, suppose that $x = 2.2$ while some numerical method produced an approximate value $\hat{x} = 2.20345$. Then the absolute error is

$$|x - \hat{x}| = 0.00345 = 3.45 \times 10^{-3}.$$

This seems quite reasonable — we have a fairly accurate solution with three correct digits and the absolute error is fairly small, on the order of 10^{-3}. We

might be very pleased with an alternative method that produced an error of 10^{-6} and horrified with a method that produced an error of 10^6.

But note that our notion of what is a large error or a small error might be thrown off completely if we were to choose a different set of units for measuring x. For example, suppose the x discussed above were measured in meters, so $x = 2.2$ meters is the correct solution. But suppose that instead we expressed the solution (and the approximate solution) in nanometers rather than meters. Then the true solution is $x = 2.2 \times 10^9$ and the approximate solution is $\hat{x} = 2.20345 \times 10^9$, giving an absolute error of

$$|x - \hat{x}| = 3.45 \times 10^6.$$

We have an error that seems huge and yet the solution is just as accurate as before, with three correct digits.

Conversely, if we measured x in kilometers, then $x = 2.2 \times 10^{-3}$ and $\hat{x} = 2.20345 \times 10^{-3}$, so

$$|x - \hat{x}| = 3.45 \times 10^{-6}.$$

The absolute error may seem much smaller, but there are still only three correct digits in the approximation.

5.3.2 Relative error

The above difficulties arise from a poor choice of scaling of the problem. One way to avoid this is to consider the *relative* error, defined by

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{x}\|}.$$

The size of the error is scaled by the size of the value being computed.

For the above examples, the relative error in $\hat{x}$ is equal to

$$\frac{|2.2 - 2.20345|}{|2.2|} = \frac{|(2.2 - 2.20345) \times 10^9|}{|2.2 \times 10^9|} = 1.57 \times 10^{-3}.$$

regardless of what units we choose. The relative error is "dimensionless" and its value is always independent of the units chosen to measure x, a very desirable feature.

Moreover, in general a relative error that is approximately 10^{-k} indicates that there are roughly k correct digits in the solution, matching our intuition.

For these reasons the relative error is often a better measure of accuracy than the absolute error. Of course if we know that our problem is "properly scaled", so that the solution $\mathbf{x}$ has a magnitude that is order 1, then it is fine to use the absolute error, which is roughly the same as the relative error in this case.

In fact it is generally better to insure that the problem is properly scaled than to rely on the relative error. Poorly scaled problems can lead to other numerical difficulties, particularly if several different scales arise in the same

problem so that some numbers are orders of magnitude larger than others for nonphysical reasons. Unless otherwise noted, we will generally assume that the problem is scaled in such a way that the absolute error is meaningful.

5.4 Conditioning

Conditioning is a fundamental issue of numerical analysis that, until now, we have skirted. It pertains to the perturbation behavior of a mathematical problem. A *well-conditioned* problem is one with the property that all small perturbations of the input lead to only small changes in the ouput of the problem. An *ill-conditioned* problem is one with the property that some small perturbation of the input can lead to a large change in the ouput.

For example, data errors form a source of perturbations. A mathematical model of some real world phenomenon typically involves some parameters or other data describing the particular situation being modeled. These values are almost never known exactly. There may be measurement errors or there may be values that cannot be measured in any precise way and must be "guessed" by the modeler. Errors in the data mean that even if the model is very good and the equation is then solved exactly it may not match reality. It is important for the numerical analyst to evaluate the effect of these data errors.

Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 10^{-10} \end{bmatrix}.$$

Its inverse matrix $\mathbf{A}^{-1}$ is

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 10^{10} \end{bmatrix}.$$

We have the product

$$\mathbf{A}\mathbf{x} = \mathbf{A} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \mathbf{b}.$$

Applying the matrix $\mathbf{A}$ to the perturbed vector $\tilde{\mathbf{x}}$ gives

$$\mathbf{A}\tilde{\mathbf{x}} = \mathbf{A} \begin{bmatrix} 1 \\ 10^{-6} \end{bmatrix} = \begin{bmatrix} 1 \\ 10^{4} \end{bmatrix} = \tilde{\mathbf{b}}.$$

Using the ∞-norm, the relative change in outputs is

$$\frac{\|\tilde{\mathbf{b}} - \mathbf{b}\|_\infty}{\|\mathbf{b}\|_\infty} = 10^4,$$

while the relative change in inputs is

$$\frac{\|\tilde{\mathbf{x}} - \mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty} = 10^{-6}.$$

Here a relatively small variation in the inputs can lead to large variations in the output vector $\mathbf{b}$. The matrix-vector product with $\mathbf{A}$ is ill-conditioned.

Consider the matrix

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1+10^{-10} & 1-10^{-10} \end{bmatrix}.$$

Its inverse matrix $\mathbf{A}^{-1}$ is

$$\mathbf{A}^{-1} = \begin{bmatrix} 1-10^{10} & 10^{10} \\ 1+10^{10} & -10^{10} \end{bmatrix}.$$

The linear system

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

has the solution

$$\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The linear system

$$\mathbf{A}\tilde{\mathbf{x}} = \begin{bmatrix} 1.1 \\ 0.9 \end{bmatrix}$$

has the solution

$$\tilde{\mathbf{x}} = \begin{bmatrix} 1.1 - 0.2 \times 10^{10} \\ 1.1 + 0.2 \times 10^{10} \end{bmatrix}.$$

Here a relatively small variation in the right hand side leads to extremely large variations in the vector $\mathbf{x}$. Solving a linear system with the matrix $\mathbf{A}$ is an ill-conditioned problem.

In the following, we define a number called to the condition number to assess what is the conditioning of a problem.

5.4.1 Error when computing the product $\mathbf{Ax} = \mathbf{b}$

Suppose that the vector $\mathbf{x}$ is perturbed to obtain the vector $\tilde{\mathbf{x}}$. If $\mathbf{b}$ and $\tilde{\mathbf{b}}$ are, respectively, satisfying $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, we would like to bound the difference between $\mathbf{b}$ and $\tilde{\mathbf{b}}$.

First, we bound the difference in absolute terms. We have

$$\begin{aligned} \|\mathbf{b} - \tilde{\mathbf{b}}\|_2 &= \|\mathbf{A}\mathbf{x} - \mathbf{A}\tilde{\mathbf{x}}\|_2 \\ \|\mathbf{b} - \tilde{\mathbf{b}}\|_2 &= \|\mathbf{A}(\mathbf{x} - \tilde{\mathbf{x}})\|_2 \end{aligned}$$

So we obtain

$$\|\mathbf{b} - \tilde{\mathbf{b}}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x} - \tilde{\mathbf{x}}\|_2. \tag{5.22}$$

This bound gives an absolute measure of the perturbation in the right hand side $\mathbf{b}$. A small norm $\|\mathbf{A}\|_2$ means that a small perturbation in the vector $\mathbf{x}$ leads to a small perturbation in the vector $\mathbf{b}$. On the other hand, a large value for $\|\mathbf{A}\|_2$ means that the perturbation in $\mathbf{b}$ can be large, even when $\|\mathbf{x} - \tilde{\mathbf{x}}\|_2$ is small.

Note that the bound (5.22) is sharp because there exists a perturbation $\mathbf{x} - \tilde{\mathbf{x}}$ such that $\|\mathbf{b} - \tilde{\mathbf{b}}\|_2 = \|\mathbf{A}\|_2 \|\mathbf{x} - \tilde{\mathbf{x}}\|_2$.

To estimate the relative perturbation, we write

$$\begin{aligned} \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2} &= \frac{\|\mathbf{A}\mathbf{x} - \mathbf{A}\tilde{\mathbf{x}}\|_2}{\|\mathbf{A}\mathbf{x}\|_2} \\ \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2} &\leq \|\mathbf{A}\|_2 \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_2}{\|\mathbf{A}\mathbf{x}\|_2} \end{aligned}$$

Next, we bound the norm of $\mathbf{A}\mathbf{x}$ with the norm of $\mathbf{A}^{-1}$ as follows

$$\|\mathbf{x}\|_2 = \|\mathbf{A}^{-1}\mathbf{A}\mathbf{x}\|_2 \leq \|\mathbf{A}^{-1}\|_2 \|\mathbf{A}\mathbf{x}\|_2 \quad \Rightarrow \quad \frac{1}{\|\mathbf{A}\mathbf{x}\|_2} \leq \frac{\|\mathbf{A}^{-1}\|_2}{\|\mathbf{x}\|_2}.$$

So we obtain

$$\frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2} \leq \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2 \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2}. \tag{5.23}$$

Definition 151. The condition number of $\mathbf{A}$ for the 2-norm is

$$\kappa_2(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2. \tag{5.24}$$

The bound (5.23) gives a relative measure of the perturbation in the right hand side $\mathbf{b}$. A small condition number means that a small perturbation in the vector $\mathbf{x}$ leads to a small perturbation in the vector $\mathbf{b}$. On the other hand, a large value for $\kappa_2(\mathbf{A})$ means that the perturbation in $\mathbf{b}$ can be large, even when $\|\mathbf{x} - \tilde{\mathbf{x}}\|_2$ is relatively small. Note that the bound (5.23) is sharp because there exists a perturbation $\mathbf{x} - \tilde{\mathbf{x}}$ such that the equality is reached.

Suppose that the matrix $\mathbf{A}$ is perturbed to obtain the matrix $\tilde{\mathbf{A}}$. If $\mathbf{b}$ and $\tilde{\mathbf{b}}$ are, respectively, satisfying $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}}$, we would like to bound the difference between $\mathbf{b}$ and $\mathbf{b}$.

First, we bound the difference in absolute terms. We have

$$\begin{aligned} \|\mathbf{b} - \tilde{\mathbf{b}}\|_2 &= \|\mathbf{A}\mathbf{x} - \tilde{\mathbf{A}}\mathbf{x}\|_2 \\ \|\mathbf{b} - \tilde{\mathbf{b}}\|_2 &= \|(\mathbf{A} - \tilde{\mathbf{A}})\mathbf{x}\|_2 \end{aligned}$$

So we obtain

$$\|\mathbf{b} - \tilde{\mathbf{b}}\|_2 \leq \|\mathbf{A} - \tilde{\mathbf{A}}\|_2 \|\mathbf{x}\|_2.$$

This bound gives an absolute measure of the perturbation in the right hand side $\mathbf{b}$.

To estimate the relative perturbation, we write

$$\begin{aligned} \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2} &= \frac{\|\mathbf{A}\mathbf{x} - \tilde{\mathbf{A}}\mathbf{x}\|_2}{\|\mathbf{A}\mathbf{x}\|_2} \\ \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2} &\leq \|\mathbf{x}\|_2 \frac{\|\mathbf{A} - \tilde{\mathbf{A}}\|_2}{\|\mathbf{A}\mathbf{x}\|_2} \end{aligned}$$

Next, we bound the norm of $\mathbf{Ax}$ with the norm of $\mathbf{A}^{-1}$ as follows

$$\frac{1}{\|\mathbf{Ax}\|_2} \leq \frac{\|\mathbf{A}^{-1}\|_2}{\|\mathbf{x}\|_2}.$$

So we obtain

$$\frac{\|\mathbf{b}-\tilde{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2} \leq \|\mathbf{A}-\tilde{\mathbf{A}}\|_2 \frac{\|\mathbf{A}^{-1}\|_2\|\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$$

and, by multiplying and dividing by $\|\mathbf{A}\|_2$,

$$\frac{\|\mathbf{b}-\tilde{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2} \leq \|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2 \frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|_2}{\|\mathbf{A}\|_2}. \tag{5.25}$$

The condition number indicates how much the vector $\mathbf{b}$ can be perturbed. A small condition number means that a small perturbation in the matrix $\mathbf{A}$ leads to a small perturbation in the vector $\mathbf{b}$. On the other hand, a large value for $\kappa_2(\mathbf{A})$ means that the perturbation in $\mathbf{b}$ can be large, even when $\|\mathbf{A}-\tilde{\mathbf{A}}\|_2$ is relatively small.

Suppose that the matrix $\mathbf{A}$ and the vector $\mathbf{x}$ are perturbed to obtain the matrix $\tilde{\mathbf{A}}$ and the vector $\tilde{\mathbf{x}}$. If $\mathbf{b}$ and $\tilde{\mathbf{b}}$ are, respectively, satisfying $\mathbf{Ax}=\mathbf{b}$ and $\tilde{\mathbf{A}}\tilde{\mathbf{x}}=\tilde{\mathbf{b}}$, we bound the difference between $\mathbf{b}$ and $\tilde{\mathbf{b}}$. The relative change satisfies

$$\frac{\|\mathbf{b}-\tilde{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2} \leq \|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2 \left(\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} + \frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|_2}{\|\mathbf{A}\|_2} + \frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|_2}{\|\mathbf{A}\|_2}\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \right) \tag{5.26}$$

Again, the condition number of $\mathbf{A}$ is the relative change magnification factor.

The condition number $\kappa_2(\mathbf{A})$ depends on the induced 2-norms of matrices $\mathbf{A}$ and $\mathbf{A}^{-1}$. For any other induced matrix norm (like the 1-norm $\|\cdot\|_1$ or the ∞-norm $\|\cdot\|_\infty$), a condition number can be similarly defined, for example:

$$\kappa_1(\mathbf{A}) = \|\mathbf{A}\|_1\|\mathbf{A}^{-1}\|_1 \quad \text{and} \quad \kappa_\infty(\mathbf{A}) = \|\mathbf{A}\|_\infty\|\mathbf{A}^{-1}\|_\infty.$$

The actual numerical value of $\kappa(\mathbf{A})$ depends on the norm being used. However, we are usually only interested in order of magnitude estimates of the condition number. So the particular norm is usually not very important.

Example 152. Consider the matrix

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1+10^{-10} & 1-10^{-10} \end{bmatrix}.$$

Its inverse matrix $\mathbf{A}^{-1}$ is

$$\mathbf{A}^{-1} = \begin{bmatrix} 1-10^{10} & 10^{10} \\ 1+10^{10} & -10^{10} \end{bmatrix}.$$

The condition number of $\mathbf{A}$ is equal to

$$\kappa_1(\mathbf{A}) = \|\mathbf{A}\|_1\|\mathbf{A}^{-1}\|_1 = \left(1 + \frac{10^{-10}}{2}\right) \times \left(2 \times 10^{10}\right) = 2 \times 10^{10} + 1.$$

For this matrix, we have also

$$\kappa_\infty(\mathbf{A}) = \|\mathbf{A}\|_\infty\|\mathbf{A}^{-1}\|_\infty = 2 \times 10^{10} + 1$$

and MATLAB gives

$$\kappa_2(\mathbf{A}) = \|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2 = 2 \times 10^{10}.$$

Applying the vector $\mathbf{x} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ to the matrix $\mathbf{A}$ results in the vector $\mathbf{b}$:

$$\mathbf{b} = \mathbf{A}\mathbf{x} = \begin{bmatrix} 0 \\ -10^{-10} \end{bmatrix}.$$

The perturbed vector $\tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ gives rise to the vector $\tilde{\mathbf{b}}$:

$$\tilde{\mathbf{b}} = \mathbf{A}\tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We have the following relative changes:

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_1}{\|\mathbf{x}\|_1} = 1 \quad \text{and} \quad \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|_1}{\|\mathbf{b}\|_1} = 2 \times 10^{10} + 1,$$

illustrating that the condition number is the magnification factor.

Note that some small perturbation of $\mathbf{x}$ can still result in small perturbation of $\mathbf{b}$. For example, the vectors

$$\mathbf{b} = \mathbf{A}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{b}} = \mathbf{A}\begin{bmatrix} 1.1 \\ 0.9 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 + 10^{-11} \end{bmatrix}$$

give the following relative changes

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_1}{\|\mathbf{x}\|_1} = 0.1 \quad \text{and} \quad \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|_1}{\|\mathbf{b}\|_1} = \frac{10^{-11}}{2}.$$

Remark. The induced norm of $\mathbf{A}$ is the largest magnification factor in a direction:

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

We can also define the smallest magnification factor:

$$m = \min_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}.$$

Note that if matrix $\mathbf{A}$ is singular, then $m = 0$. The condition number is sometimes defined as the ratio

$$\kappa(\mathbf{A}) = \frac{\|\mathbf{A}\|}{m}. \tag{5.27}$$

Expressions (5.24) and (5.27) are equal since

$$m = \min_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} = \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{x}\|}{\|\mathbf{A}\mathbf{x}\|} = \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}^{-1}(\mathbf{A}\mathbf{x})\|}{\|\mathbf{A}\mathbf{x}\|} = \|\mathbf{A}^{-1}\|.$$

For any p-norm, we have

$$\kappa_p(\mathbf{A}) \geq 1. \tag{5.28}$$

In the 2-norm, orthogonal matrices are perfectly conditioned in that $\kappa(\mathbf{Q}) = 1$ if $\mathbf{Q}$ is orthogonal.

5.4.2 Error when solving the system $\mathbf{Ax} = \mathbf{b}$

Given an invertible matrix $\mathbf{A}$ and the vector $\mathbf{b}$, solving the system $\mathbf{Ax} = \mathbf{b}$ is mathematically identical to computing the vector $\mathbf{A}^{-1}\mathbf{b}$. This is equivalent to the problem just studied, except that the matrix $\mathbf{A}$ is replaced by the matrix $\mathbf{A}^{-1}$. The condition numbers $\kappa(\mathbf{A})$ and $\kappa(\mathbf{A}^{-1})$ are equal since $\left(\mathbf{A}^{-1}\right)^{-1} = \mathbf{A}$.

For a rectangular matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $(m \geq n)$, of full rank, the condition number is defined in terms of the pseudoinverse

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{\dagger}\| = \|\mathbf{A}\|\|(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\|.$$

These definitions extend also to complex matrices.

The size of the condition number is a good measure of how close to singular the matrix is. It is finite as long as $\mathbf{A}$ is nonsingular, but approaches ∞ as $\mathbf{A}$ approaches a singular matrix. In particular, the condition number gives a good indication of how much accuracy we can expect in a computed solution to $\mathbf{Ax} = \mathbf{b}$. When solving a problem $\mathbf{Ax} = \mathbf{b}$ with a fixed digits of precision, one must always expect to "lose" $\log_{10}\kappa(\mathbf{A})$ digits of accuracy in computing the solution.

Example 153. Consider the matrix and vectors

$$\mathbf{A} = \begin{bmatrix} 4.1 & 2.8 \\ 9.7 & 6.6 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 4.1 \\ 9.7 \end{bmatrix}.$$

Clearly, $\mathbf{Ax} = \mathbf{b}$ and we have

$$\|\mathbf{x}\|_1 = 1 \quad \text{and} \quad \|\mathbf{b}\| = 13.8.$$

If the right-hand side is changed to

$$\tilde{\mathbf{b}} = \begin{bmatrix} 4.11 \\ 9.70 \end{bmatrix},$$

the solution becomes

$$\tilde{\mathbf{x}} = \begin{bmatrix} 0.34 \\ 0.97 \end{bmatrix}.$$

The relative changes

$$\frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|_1}{\|\mathbf{b}\|_1} = 0.0007246 \quad \text{and} \quad \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_1}{\|\mathbf{x}\|_1} = 1.63.$$

$\kappa(\mathbf{A})$ being the magnification factor, we have

$$\kappa_1(\mathbf{A}) \geq \frac{1.63}{0.0007246} = 2249.4.$$

For this example, we have actually an equality

$$\kappa_1(\mathbf{A}) = 2249.4.$$

Example 154. The Hilbert matrix $\mathbf{H}_n = (h_{ij})_{n\times n}$ is defined by

$$h_{ij} = \frac{1}{i+j-1}.$$

This matrix is often used for test purposes because of its ill-conditioned nature. The condition number $\kappa(\mathbf{H}_n)$ increases rapidly with n and can grow like $ce^{3.5n}$.

5.4.3 Error when evaluating a general function

Consider the function $f : \mathbb{R} \longrightarrow \mathbb{R}$ that is differentiable. The function f is usually nonlinear.

When evaluating the function f at x, we can again compute the absolute error on $f(x)$ if x is slightly perturbed:

$$f(x+h) - f(x) = f'(\xi)h \approx hf'(x)$$

where we invoked the mean-value theorem.

On the other hand, the relative error is

$$\frac{f(x+h) - f(x)}{f(x)} \approx \frac{hf'(x)}{f(x)} = \left[\frac{xf'(x)}{f(x)}\right] \frac{x+h-x}{x}.$$

So the factor $xf'(x)/f(x)$ is the condition number for this problem.

The condition number can be extended to general functions. If the differentiable function f is from a linear space U equipped with a norm into a linear space V equipped with a norm, the absolute error satisfies

$$\|f(u) - f(\tilde{u})\| \approx \|J(u)(u-\tilde{u})\| \leq \|J(u)\|\|u-\tilde{u}\|.$$

The relative error is

$$\frac{\|f(u) - f(\tilde{u})\|}{\|f(u)\|} \approx \frac{\|J(u)(u-\tilde{u})\|}{\|f(u)\|} \leq \frac{\|J(u)\|}{\|f(u)\|}\|u-\tilde{u}\| = \left[\frac{\|J(u)\|\|u\|}{\|f(u)\|}\right] \frac{\|u-\tilde{u}\|}{\|u\|}.$$

The condition number becomes

$$\frac{\|J(u)\|\|u\|}{\|f(u)\|}.$$

Example 155. Consider the problem of computing $\sqrt{x}$ for $x > 0$. The Jacobian of $f : x \mapsto \sqrt{x}$ is the derivative $f'(x) = 1/(2\sqrt{x})$. So we have

$$\kappa = \frac{xf'(x)}{f(x)} = \frac{1/(2\sqrt{x})}{\sqrt{x}/x} = \frac{1}{2}.$$

This is a well-conditioned problem.

Example 156. Consider the problem

$$f : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \longmapsto x_1 - x_2.$$

The Jacobian of f is

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 & -1 \end{bmatrix}.$$

Using the ∞-norm, we get

$$\kappa = \frac{\|J(\mathbf{x})\|_\infty \|\mathbf{x}\|_\infty}{|f(\mathbf{x})|} = \frac{2\max(|x_1|, |x_2|)}{|x_1 - x_2|}.$$

This quantity is large when $|x_1 - x_2| \approx 0$. So the problem is ill-conditioned when $x_1 \approx x_2$. This example highlights the hazards of cancellation error.

5.5 Useful commands in MATLAB

Here are a few commands in MATLAB useful for this section.

- `norm(A)` computes the induced 2-norm of matrix **A**. `norm(A,inf)`, `norm(A,1)`, `norm(A,p)`, `norm(A,'fro')` computes, respectively, the ∞-norm, the 1-norm, the p-norm, and the Frobenius norm.
- `cond(A)` computes the matrix condition number in 2-norm. `cond(A,p)` returns the matrix condition number in p-norm: `norm(A,p)*norm(inv(A),p)`.

5.6 Exercises

Exercise 157. Prove that the norm $\|\cdot\|_{(m,n)}$ satisfies the properties (5.1).

Exercise 158. Prove that, for any $m \times n$ matrix, we have

$$\frac{1}{n}\|\mathbf{A}\|_\infty \leq \|\mathbf{A}\|_1 \leq m\|\mathbf{A}\|_\infty.$$

Exercise 159. Prove that, for any $m \times n$ matrix, we have

$$\frac{1}{\sqrt{m}}\|\mathbf{A}\|_1 \leq \|\mathbf{A}\|_2 \leq \sqrt{n}\|\mathbf{A}\|_1.$$

Exercise 160. Prove that, for any $m \times n$ matrix, we have

$$\|\mathbf{A}\|_2 \leq \|\mathbf{A}\|_F \leq \sqrt{n}\|\mathbf{A}\|_2.$$

Exercise 161. Consider $U = \mathbb{R}^n$ and $\mathbf{W}$ a nonsingular matrix. Show that the map

$$\|\mathbf{x}\|_{\mathbf{W}} = \|\mathbf{W}\mathbf{x}\|_2$$

is a norm.

Exercise 162. Consider a real square matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ such that $\mathbf{M}$ is symmetric and of full rank. $\mathbf{M}$ is also positive definite, *i.e* it safisfies

$$\forall \mathbf{x} \in \mathbb{R}^n, \text{ if } \mathbf{x} \neq \mathbf{0}, \ \mathbf{x}^T\mathbf{M}\mathbf{x} > 0.$$

- Show that, the map
$$(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{x}^T\mathbf{M}\mathbf{y}$$
is an inner product in $\mathbb{R}^n$. Verify all the steps in the definition of an inner product.
- Write the Cauchy-Schwarz inequality for this inner product.

Exercise 163. Consider $U = C^0([0,1], \mathbb{R})$. Show that the map

$$f \longmapsto \sqrt{\int_0^1 f(t)^2 dt}$$

is a norm.

Exercise 164. Prove that an inner product $< \cdot, \cdot >$ on a real linear space U satisfies

$$\forall u, v, w \in U, \quad \forall \alpha, \beta \in \mathbb{R}, \quad < u, \alpha v + \beta w >= \alpha < u, v > + \beta < u, w > . \quad (5.29)$$

Exercise 165. Prove the Cauchy-Schwarz inequality (5.16).

Exercise 166. Prove Proposition 147.

Exercise 167. Consider $U = C^0([0,1], \mathbb{R})$. Show that the map

$$(f, g) \longmapsto \int_0^1 f(t)g(t)dt$$

is an inner product.

Exercise 168. Prove the bound (5.26).

Exercise 169. Consider the MATLAB function

```
function t = test(A)
m = size(A, 1);
n = size(A, 2);
t = 0.0;
for ii = 1:m,
  for jj = 1:n,
    if abs(A(ii,jj)) > t
      t = abs(A(ii,jj));
    end
  end
end
```

- Write the mathematical expression for this function.
- Check whether the function is a norm.
- For the case where $m = n$, does this norm have the property $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$?

6 The QR factorization

The QR algorithm is a very important algorithmic idea in numerical linear algebra.

6.1 Reduced QR factorization

For many applications, we are interested in the column span of a matrix $\mathbf{A}$ and in the knowledge of a basis for this subspace. The idea of QR factorization is the construction of orthonormal vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots$ that span the range of $\mathbf{A}$.

Assume that $\mathbf{A} \in \mathbb{R}^{m\times n}$ $(m \geq n)$ has full rank n and denote the column vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$. We want to build the sequence of orthonormal vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ such that

$$\text{span}(\mathbf{a}_1, \ldots, \mathbf{a}_j) = \text{span}(\mathbf{q}_1, \ldots, \mathbf{q}_j) \qquad j = 1, \ldots, n. \tag{6.1}$$

When $j = 1$, the relation (6.1) takes the form

$$\mathbf{a}_1 = \mathbf{q}_1 r_{11} \quad \text{such that} \quad \|\mathbf{q}_1\|_2 = 1, \tag{6.2}$$

which defines the first vector $\mathbf{q}_1$ and a coefficient r_{11}

$$\|\mathbf{a}_1\|_2 = \|\mathbf{q}_1 r_{11}\| = |r_{11}|\|\mathbf{q}_1\| = |r_{11}|.$$

Arbitrarily, we may choose that $r_{11} > 0$. Note that r_{11} will not be zero because the matrix $\mathbf{A}$ has full rank.

When $j = 2$ and the first vector $\mathbf{q}_1$ is defined, the relation (6.1) becomes

$$\mathbf{a}_2 = \mathbf{q}_1 r_{12} + \mathbf{q}_2 r_{22} \quad \text{such that} \quad \mathbf{q}_1^T \mathbf{q}_2 = 0 \quad \text{and} \quad \|\mathbf{q}_2\|_2 = 1, \tag{6.3}$$

which implies

$$\mathbf{q}_1^T \mathbf{a}_2 = \mathbf{q}_1^T \mathbf{q}_1 r_{12} = r_{12}$$

and

$$\|\mathbf{a}_2 - \mathbf{q}_1 r_{12}\|_2 = \|\mathbf{q}_2 r_{22}\| = |r_{22}|.$$

r_{22} will not be zero else the column vector $\mathbf{a}_2$ would be linearly dependent with $\mathbf{a}_1$. It is impossibe because the matrix $\mathbf{A}$ has full rank. We can choose again that $r_{22} > 0$.

When $j = 3$ and the vectors $\mathbf{q}_1$ and $\mathbf{q}_2$ are defined, the relation (6.1) becomes

$$\mathbf{a}_3 = \mathbf{q}_1 r_{13} + \mathbf{q}_2 r_{23} + \mathbf{q}_3 r_{33} \text{ such that } \mathbf{q}_1^T \mathbf{q}_3 = 0,\ \mathbf{q}_2^T \mathbf{q}_3 = 0, \text{ and } \|\mathbf{q}_3\|_2 = 1, \tag{6.4}$$

which implies

$$\mathbf{q}_1^T \mathbf{a}_3 = \mathbf{q}_1^T \mathbf{q}_1 r_{13} = r_{13} \ \text{ and } \ \mathbf{q}_2^T \mathbf{a}_3 = \mathbf{q}_2^T \mathbf{q}_2 r_{23} = r_{23}$$

and

$$\|\mathbf{a}_3 - \mathbf{q}_1 r_{13} - \mathbf{q}_2 r_{23}\|_2 = \|\mathbf{q}_3 r_{33}\| = |r_{33}|.$$

r_{33} will not be zero else the column vector $\mathbf{a}_3$ would be linearly dependent with $(\mathbf{a}_1, \mathbf{a}_2)$. It is impossibe because the matrix $\mathbf{A}$ has full rank. We can choose again that $r_{33} > 0$.

Generally, when $1 \leq j \leq n$ and the $j-1$ orthonormal vectors $\mathbf{q}_1, \ldots, \mathbf{q}_{j-1}$ are defined, we write

$$\mathbf{a}_j = \mathbf{q}_1 r_{1j} + \ldots + \mathbf{q}_j r_{jj}. \tag{6.5}$$

It is evident that an appropriate definition of the coefficients of r_{ij} is

$$r_{ij} = \mathbf{q}_i^T \mathbf{a}_j \tag{6.6}$$

and

$$r_{jj} = \|\mathbf{a}_j - \sum_{i=1}^{j-1} \mathbf{q}_i (\mathbf{q}_i^T \mathbf{a}_j)\|_2. \tag{6.7}$$

Relation (6.1) can be rewritten in a matrix form

$$\left[\begin{array}{c|c|c} & & \\ \mathbf{a}_1 & \cdots & \mathbf{a}_n \\ & & \end{array}\right] = \left[\begin{array}{c|c|c} & & \\ \mathbf{q}_1 & \cdots & \mathbf{q}_n \\ & & \end{array}\right] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & & r_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{nn} \end{bmatrix} \tag{6.8}$$

or

$$\mathbf{A} = \widehat{\mathbf{Q}}\widehat{\mathbf{R}}, \tag{6.9}$$

where $\widehat{\mathbf{Q}}$ is $m \times n$ with orthonormal columns and $\widehat{\mathbf{R}}$ is $n \times n$ upper-triangular matrix with positive diagonal entries. Such a factorization is called a reduced QR factorization of $\mathbf{A}$.

Theorem 170. *Each matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$ of full rank has a unique reduced QR factorization $\mathbf{A} = \widehat{\mathbf{Q}}\widehat{\mathbf{R}}$ with $r_{jj} > 0$.*

The proof is provided by the constructive method described earlier. This method is an old idea, known as the Gram-Schmidt orthogonalization. In Figure 6.1, we represent graphically the reduced QR factorization.

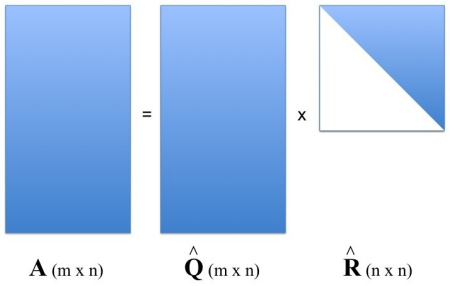


Figure 6.1: Reduced QR factorization $(m \geq n)$

Remark. Each complex matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ $(m \geq n)$ of full rank has also a unique reduced QR factorization $\mathbf{A} = \widehat{\mathbf{Q}}\widehat{\mathbf{R}}$ with $r_{jj} > 0$.

Remark. The matrix $\hat{\mathbf{Q}}$ has orthonormal columns. So we have $\hat{\mathbf{Q}}^T\hat{\mathbf{Q}} = \mathbf{I}$, where the identity matrix has dimension $n \times n$. Then we get

$$\forall x \in \mathbb{R}^{n\times 1},\ \|\hat{\mathbf{Q}}\mathbf{x}\|_2 = \sqrt{\left(\hat{\mathbf{Q}}\mathbf{x}\right)^T \hat{\mathbf{Q}}\mathbf{x}} = \sqrt{\mathbf{x}^T\hat{\mathbf{Q}}^T\hat{\mathbf{Q}}\mathbf{x}} = \sqrt{\mathbf{x}^T\mathbf{x}} = \|\mathbf{x}\|_2$$

which indicates that $\|\hat{\mathbf{Q}}\|_2 = 1$. So we obtain

$$\|\mathbf{A}\|_2 = \|\hat{\mathbf{R}}\|_2.$$

Example 171. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 1 & 2 \end{bmatrix}.$$

We will compute its QR factorization. The matrix $\mathbf{A}$ has a determinant equal to 4 and it is of full rank. The first step constructs the vector $\mathbf{q}_1$ such that

$$\mathbf{a}_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \quad \|\mathbf{a}_1\|_2 = \sqrt{10} \quad \Longrightarrow \mathbf{q}_1 = \begin{bmatrix} 3/\sqrt{10} \\ 1/\sqrt{10} \end{bmatrix}.$$

Next, we write

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \frac{r_{12}}{\sqrt{10}} + \mathbf{q}_2 r_{22} \quad \text{such that} \quad \mathbf{q}_1^T\mathbf{q}_2 = 0 \quad \text{and} \quad \|\mathbf{q}_2\|_2 = 1,$$

which gives

$$\begin{bmatrix} 3/\sqrt{10} \\ 1/\sqrt{10} \end{bmatrix}^T \begin{bmatrix} 2 \\ 2 \end{bmatrix} = r_{12} \quad \Longrightarrow \quad r_{12} = \frac{8}{\sqrt{10}}.$$

So we obtain

$$\begin{aligned} \mathbf{q}_2 r_{22} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \frac{8}{\sqrt{10}} \begin{bmatrix} 3/\sqrt{10} \\ 1/\sqrt{10} \end{bmatrix} = \begin{bmatrix} 2 - \frac{24}{10} \\ 2 - \frac{8}{10} \end{bmatrix} = \begin{bmatrix} -\frac{4}{10} \\ \frac{12}{10} \end{bmatrix} \\ = \begin{bmatrix} -\frac{4}{\sqrt{160}} \\ \frac{12}{\sqrt{160}} \end{bmatrix} \frac{\sqrt{160}}{10} = \begin{bmatrix} -\frac{1}{\sqrt{10}} \\ \frac{3}{\sqrt{10}} \end{bmatrix} \frac{2\sqrt{10}}{5}. \end{aligned}$$

Finally, we write

$$\begin{bmatrix} 3 & 2 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} \frac{3}{\sqrt{10}} & \frac{-1}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} & \frac{3}{\sqrt{10}} \end{bmatrix} \begin{bmatrix} \sqrt{10} & \frac{8}{\sqrt{10}} \\ 0 & \frac{2\sqrt{10}}{5} \end{bmatrix}.$$

Example 172. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 3 & 0 \\ 1 & 3 & 6 \end{bmatrix}.$$

We will compute its QR factorization. The matrix $\mathbf{A}$ is of full rank. The first step constructs the vector $\mathbf{q}_1$ such that

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \|\mathbf{a}_1\|_2 = \sqrt{3} \quad \Longrightarrow \mathbf{q}_1 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Next, we write

$$\begin{bmatrix} 0 \\ 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{r_{12}}{\sqrt{3}} + \mathbf{q}_2 r_{22} \quad \text{such that} \quad \mathbf{q}_1^T \mathbf{q}_2 = 0 \quad \text{and} \quad \|\mathbf{q}_2\|_2 = 1,$$

which gives

$$\left(\frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right)^T \begin{bmatrix} 0 \\ 3 \\ 3 \end{bmatrix} = r_{12} \quad \Longrightarrow \quad r_{12} = \frac{6}{\sqrt{3}} = 2\sqrt{3}.$$

Then we have

$$\mathbf{a}_2 - \mathbf{q}_1 r_{12} = \begin{bmatrix} 0 \\ 3 \\ 3 \end{bmatrix} - \frac{2\sqrt{3}}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 3 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}$$

and

$$r_{22} = \sqrt{6} \quad \text{and} \quad \mathbf{q}_2 = \frac{1}{\sqrt{6}} \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}.$$

For the third column, we write

$$\begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{r_{13}}{\sqrt{3}} + \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix} \frac{r_{23}}{\sqrt{6}} + \mathbf{q}_3 r_{33}$$
$$\text{such that} \quad \mathbf{q}_1^T \mathbf{q}_3 = 0, \ \mathbf{q}_2^T \mathbf{q}_3 = 0, \text{ and } \|\mathbf{q}_3\|_2 = 1.$$

We obtain

$$\left(\frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right)^T \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} = r_{13} \quad \Longrightarrow \quad r_{13} = 2\sqrt{3}$$

and

$$\left(\frac{1}{\sqrt{6}} \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix} \right)^T \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} = r_{23} \quad \Longrightarrow \quad r_{23} = \sqrt{6}.$$

The third column of $\hat{\mathbf{Q}}$ will satisfy

$$\mathbf{q}_3 r_{33} = \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} - \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -3 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \times \frac{1}{\sqrt{2}} \times 3\sqrt{2}.$$

Finally, we write

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 3 & 0 \\ 1 & 3 & 6 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{-2}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 2\sqrt{3} & 2\sqrt{3} \\ 0 & \sqrt{6} & \sqrt{6} \\ 0 & 0 & 3\sqrt{2} \end{bmatrix}.$$

6.2 Gram-Schmidt orthogonalization

The steps, described in (6.5), (6.6), and (6.10), form the j-th iteration of the Gram-Schmidt orthogonalization. Mathematically, it offers a simple route to understanding and proving various properties of QR factorizations. The complete algorithm is described in Algorithm 1.

Algorithm 1 Classical Gram-Schmidt

```
for j = 1 to n do
  v = a_j
  for i = 1 to j − 1 do
    r_ij = q_i^T a_j
    v = v − r_ij q_i
  end for
  r_jj = ‖v‖_2
  q_j = v/r_jj
end for
```

6.3 Projections

Equation (6.10) can be written in a matrix form

$$r_{jj} = \|(\mathbf{I} - \mathbf{Q}_{j-1}\mathbf{Q}_{j-1}^T)\mathbf{a}_j\|_2. \tag{6.10}$$

where the matrix $\mathbf{Q}_{j-1}$ denote the $m \times (j-1)$ matrix containing the first $j-1$ column vectors $\mathbf{q}_1, \ldots, \mathbf{q}_{j-1}$. The matrix

$$\mathbf{P}_j = \mathbf{I} - \mathbf{Q}_{j-1}\mathbf{Q}_{j-1}^T \tag{6.11}$$

represents an orthogonal projection. The projection $\mathbf{P}_j$ computes the component of $\mathbf{a}_j$ that is in the span of $\mathbf{Q}_{j-1}$ and removes it. Another interpretation is that the projector $\mathbf{P}_j$ computes the component of $\mathbf{a}_j$ that is not in the span of $\mathbf{Q}_{j-1}$.

Definition 173. A projection, or projector, is a square matrix $\mathbf{P}$ such that $\mathbf{P}^2 = \mathbf{P}$. It is an orthogonal projector (in contrast to an oblique projector) when $\mathbf{P}^T = \mathbf{P}$.

Trefethen and Bau[2] note that

[2] L. N. Trefethen and D. Bau, *Numerical linear algebra*, SIAM, Philadelphia, 1997.

> "The term projector might be thought of as arising from the notion that if one were to shine a light onto the range of $\mathbf{P}$ from the right direction, then $\mathbf{Pv}$ would be the shadow projected by the vector $\mathbf{v}$."

Note that we have

$$\mathbf{P}^2\mathbf{v} = \mathbf{Pv} \implies \mathbf{P}^2\mathbf{v} - \mathbf{Pv} = \mathbf{0} \implies \mathbf{P}(\mathbf{Pv} - \mathbf{v}) = \mathbf{0}. \tag{6.12}$$

This means that $\mathbf{Pv} - \mathbf{v}$ belongs to the null space of $\mathbf{P}$.

Proposition 174. *Let $\mathbf{Q}$ be an $m \times n$ matrix with orthonormal columns, i.e. $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. The matrices $\mathbf{QQ}^T$ and $\mathbf{I} - \mathbf{QQ}^T$ are orthogonal projectors.*

Proposition 175. *If $\mathbf{P}$ is a projector, then $\mathbf{I} - \mathbf{P}$ is also a projector. The matrix $\mathbf{I} - \mathbf{P}$ is called the complementary projector to $\mathbf{P}$.*

$\mathbf{I} - \mathbf{P}$ projects exactly into the nullspace of $\mathbf{P}$. If $\mathbf{v}$ belongs to the null space of $\mathbf{P}$, then $(\mathbf{I} - \mathbf{P})v = \mathbf{v}$ belongs to the range of $\mathbf{I} - \mathbf{P}$, which implies

$$\mathcal{N}(\mathbf{P}) \subset \mathcal{R}(\mathbf{I} - \mathbf{P}).$$

If $\mathbf{v}$ belongs to range$(\mathbf{I} - \mathbf{P})$, then $\mathbf{v} = (\mathbf{I} - \mathbf{P})\mathbf{v}$ and

$$\mathbf{Pv} = \mathbf{P}\left[(\mathbf{I} - \mathbf{P})\mathbf{v}\right] = \mathbf{Pv} - \mathbf{P}^2\mathbf{v} = \mathbf{0},$$

which implies that

$$\mathcal{N}(\mathbf{P}) = \mathcal{R}(\mathbf{I} - \mathbf{P}). \tag{6.13}$$

By writing $\mathbf{P} = \mathbf{I} - (\mathbf{I} - \mathbf{P})$, we derive the complementary fact

$$\mathcal{N}(\mathbf{I} - \mathbf{P}) = \mathcal{R}(\mathbf{P}). \tag{6.14}$$

We can also see that $\mathcal{R}(\mathbf{P}) \cap \mathcal{R}(\mathbf{I} - \mathbf{P}) = \{\mathbf{0}\}$: any vector $\mathbf{v}$ in both sets satisfies

$$\mathbf{Pv} = \mathbf{v} = \mathbf{v} - \mathbf{Pv} = \mathbf{0}.$$

Using the result (4.12) about dimensions, these computations show that an $m \times m$ projector $\mathbf{P}$ separates $\mathbb{R}^m$ into two spaces,

$$\mathbb{R}^m = \mathcal{R}(\mathbf{P}) \oplus \mathcal{R}(\mathbf{I} - \mathbf{P}). \tag{6.15}$$

6.4 Full QR factorization

A full QR factorization of $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$ goes further, appending an additional $m - n$ orthonormal columns to $\hat{\mathbf{Q}}$ so that it becomes an $m \times m$ orthogonal matrix $\mathbf{Q}$. In the process, row of zeros are appended to $\hat{\mathbf{R}}$ so that it becomes an $m \times n$ matrix $\mathbf{R}$, still upper-triangular. In Figure 6.2, we represent graphically the reduced QR factorization.

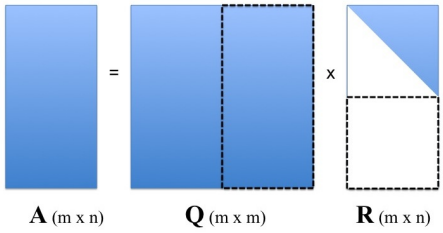


Figure 6.2: Full QR factorization ($m \geq n$)

When the matrix $\mathbf{A}$ does not have full rank, the Gram-Schmidt algorithm can break when r_{jj} is zero or $\mathbf{v} = \mathbf{0}$. In that case, we simply choose the vector $\mathbf{q}_j$ to be any arbitrary normalized vector orthogonal to the previously constructed vectors $\mathbf{q}_1, \ldots, \mathbf{q}_{j-1}$ and continue the Gram-Schmidt algorithm.

Theorem 176. *Every matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$ *has a QR factorization* $\mathbf{A} = \mathbf{QR}$.

Example 177. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 6 \end{bmatrix}.$$

We will compute its QR factorization. The matrix $\mathbf{A}$ is of rank 2. The first step constructs the vector $\mathbf{q}_1$ such that

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \|\mathbf{a}_1\|_2 = \sqrt{3} \quad \Longrightarrow \mathbf{q}_1 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Next, we write

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{r_{12}}{\sqrt{3}} + \mathbf{q}_2 r_{22} \quad \text{such that} \quad \mathbf{q}_1^T \mathbf{q}_2 = 0 \quad \text{and} \quad \|\mathbf{q}_2\|_2 = 1,$$

which gives

$$\left(\frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right)^T \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = r_{12} \quad \Longrightarrow \quad r_{12} = 0.$$

Then we have

$$\mathbf{a}_2 - \mathbf{q}_1 r_{12} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

which implies that $r_{22} = 0$. The vector $\mathbf{q}_2$ can not be zero because it is of norm 1. Next we simply choose the vector $\mathbf{q}_2$ to be any arbitrary normalized

vector orthogonal to the previously constructed vector $\mathbf{q}_1$ and continue the Gram-Schmidt algorithm. For instance, we can choose

$$\mathbf{q}_2 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}.$$

For the third column, we write

$$\begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{r_{13}}{\sqrt{3}} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \frac{r_{23}}{\sqrt{6}} + \mathbf{q}_3 r_{33}$$

$$\text{such that} \quad \mathbf{q}_1^T \mathbf{q}_3 = 0, \ \mathbf{q}_2^T \mathbf{q}_3 = 0, \text{ and } \|\mathbf{q}_3\|_2 = 1.$$

We obtain

$$\left(\frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right)^T \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} = r_{13} \quad \Longrightarrow \quad r_{13} = 2\sqrt{3}$$

and

$$\left(\frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \right)^T \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} = r_{23} \quad \Longrightarrow \quad r_{23} = \sqrt{6}.$$

The third column of $\mathbf{Q}$ will satisfy

$$\mathbf{q}_3 r_{33} = \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \times \frac{1}{\sqrt{2}} \times 3\sqrt{2}.$$

Finally, we write

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 6 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{-2}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 & 2\sqrt{3} \\ 0 & 0 & \sqrt{6} \\ 0 & 0 & 3\sqrt{2} \end{bmatrix}.$$

Note that a different choice for $\mathbf{q}_2$ would result in a different factorization. For example, setting

$$\mathbf{q}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix},$$

we obtain

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 6 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 & 2\sqrt{3} \\ 0 & 0 & 3\sqrt{2} \\ 0 & 0 & \sqrt{6} \end{bmatrix}.$$

Remark. Each complex matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$ $(m \geq n)$ has also a full QR factorization $\mathbf{A} = \mathbf{QR}$.

6.5 Solution of $\mathbf{Ax} = \mathbf{b}$ by QR factorization

Suppose we wish to solve $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A}$ is a real square nonsingular matrix. Using the QR factorization, we can write $\mathbf{QRx} = \mathbf{b}$ or

$$\mathbf{Rx} = \mathbf{Q}^T\mathbf{b} \tag{6.16}$$

The right hand side is easy to compute when $\mathbf{Q}$ is known. The system of linear equations in $\mathbf{x}$ is also easy to solve because $\mathbf{R}$ is triangular. This suggests the following method for computing the solution to $\mathbf{Ax} = \mathbf{b}$:

1. Compute a QR factorization $\mathbf{A} = \mathbf{QR}$
2. Compute $\mathbf{y} = \mathbf{Q}^T\mathbf{b}$
3. Solve $\mathbf{Rx} = \mathbf{y}$ for $\mathbf{x}$

This combination is an excellent method for solving linear systems of equations. However, it is not the standard methods for such problems. Gaussian elimination is the algorithm often used in practice, since it requires only half as many numerical operations. The Gaussian elimination will be described later in the notes.

6.6 Generalization of Gram-Schmidt orthogonalization

The Gram-Schmidt orthogonalization has an analogue for an abstract real linear space. Let U be a real linear space equipped with an inner product $< \cdot, \cdot >$ and the associated norm $\|\cdot\| = \sqrt{< \cdot, \cdot >}$. The complete algorithm is described in Algorithm 2.

Algorithm 2 Classical Gram-Schmidt with a general inner-product

Require: $(u^{(1)}, \ldots, u^{(n)})$, n members of U
  for $j = 1$ to n **do**
    $v = u^{(j)}$
    for $i = 1$ to $j - 1$ **do**
      $r_{ij} =< q^{(i)}, u^{(j)} >$
      $v = v - r_{ij}q^{(i)}$
    end for
    $r_{jj} = \|v\| = \sqrt{< v, v >}$
    $q^{(j)} = v/r_{jj}$
  end for

6.7 Useful commands in MATLAB

Here are a few commands in MATLAB useful for this section.

- `[Q,R]=qr(A,0)` computes the reduced QR factorization of $\mathbf{A}$.
- `[Q,R]=qr(A)` computes the full QR factorization of $\mathbf{A}$.

6.8 Exercises

Exercise 178. Prove Proposition 174.

Exercise 179. Consider $\mathcal{P}_3$ the space of polynomials of degree at most 3, equipped with the inner product

$$< f, g >= \int_{-1}^{1} f(t)g(t)dt.$$

Starting from the basis $(1, x, x^2, x^3)$, use the Gram-Schmidt construction to build an orthonormal basis. The resulting polynomials are scalar multiples of what are known as the Legendre polynomials, P_j, which are conventionally normalized so that $P_j(1) = 1$. Computations with such polynomials form the basis of spectral methods, one of the most powerful techniques for the numerical solution of partial differential equations.

Exercise 180. Compute a QR factorization for the following matrices:

1. $\mathbf{A} = \begin{bmatrix} 1 & 6 & 14 \\ 1 & -2 & -8 \\ 1 & 6 & 2 \\ 1 & -2 & 4 \end{bmatrix}$ with $\mathbf{Q} \in \mathbb{R}^{4\times 3}$ and $r_{ii} > 0$.
2. $\mathbf{B} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$ with $r_{ii} \geq 0$.

Exercise 181. Compute, by hand, the QR factorization for the matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 0 \\ 2 & 1 & 4 \\ 0 & 4 & 5 \end{bmatrix}$$

7 Computer arithmetic

Computer memory is partitioned into bytes consisting of 8 bits. Numbers are usually stored in the computer using either 4 bytes (32 bits, sometimes called single precision) or 8 bytes (64 bits, or double precision). MATLAB always uses 64-bit floating point numbers (or double precision). Consequently, computers can only represent a finite subset of real numbers. It is good to be aware of this limitation and of potential pitfalls of computer arithmetic. Careless construction of algorithms can have sometimes dramatic consequences as we will see in some examples.

7.1 Digital representation limitations

Represented numbers in computer arithmetic can not be arbitrarily large or small. For example, the IEEE double precision arithmetic permits numbers as large as 1.79×10^{308} and as small as 2.23×10^{-308}. This range is usually sufficiently large. Overflow and underflow are rarely a serious hazard.

Represented numbers have gaps between them. For example, in IEEE double precision arithmetic, the interval $[1, 2]$ is represented by the discrete subset

$$1,\ 1 + 2^{-52},\ 1 + 2 \times 2^{-52},\ 1 + 3 \times 2^{-52},\ \ldots,\ 2. \tag{7.1}$$

The interval $[2, 4]$ is represented by the discrete subset

$$2,\ 2 + 2^{-51},\ 2 + 2 \times 2^{-51},\ 2 + 3 \times 2^{-51},\ \ldots,\ 4,$$

which are the same numbers multiplied by 2. In general, the interval $[2^j, 2^{j+1}]$ is represented by the numbers in (7.1) multiplied by 2^j. In IEEE double precision arithmetic, the relative gap between two numbers is never larger than $2^{52} \approx 2.2 \times 10^{-16}$. This gap is usually negligible but ill-conditioned problems or unstable algorithms can be affected by this finite gap.

Example 182. Consider the matrix

$$\mathbf{A}_\varepsilon = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1+\varepsilon & 1-\varepsilon \end{bmatrix}$$

and its inverse

$$\mathbf{A}_\varepsilon^{-1} = \begin{bmatrix} 1 - 1/\varepsilon & 1/\varepsilon \\ 1 + 1/\varepsilon & -1/\varepsilon \end{bmatrix}$$

where $0 < \varepsilon < 1$. The 1-norms for these matrices are

$$\|\mathbf{A}_\varepsilon\|_1 = 1 + \frac{\varepsilon}{2} \quad \|\mathbf{A}_\varepsilon^{-1}\|_1 = \frac{2}{\varepsilon}$$

and the condition number is

$$\kappa_1(\mathbf{A}_\varepsilon) = \|\mathbf{A}_\varepsilon\|_1 \|\mathbf{A}_\varepsilon^{-1}\|_1 = \left(1 + \frac{\varepsilon}{2}\right) \times \frac{2}{\varepsilon} = \frac{2}{\varepsilon} + 1.$$

Recall that the condition number gives a good indication of how much accuracy we can expect in a computed solution to $\mathbf{A}_\varepsilon \mathbf{x} = \mathbf{b}$. When solving a problem $\mathbf{A}_\varepsilon \mathbf{x} = \mathbf{b}$ with a fixed digits of precision, one must always expect to "lose" $\log_{10} \kappa_1(\mathbf{A}_\varepsilon)$ digits of accuracy in computing the solution. For any value of ε, we have

$$\mathbf{A}_\varepsilon \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We solve the same system numerically with MATLAB and compute the 1-norm of the error

	$\|\mathbf{A}_\varepsilon \backslash \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}\|_1$	$\|inv(\mathbf{A}_\varepsilon) * \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}\|_1$
$\varepsilon = 10^{-5}$	$\approx 1.11 \times 10^{-11}$	$\approx 1.455 \times 10^{-11}$
$\varepsilon = 10^{-8}$	$\approx 1.11 \times 10^{-08}$	$\approx 1.490 \times 10^{-08}$
$\varepsilon = 10^{-12}$	$\approx 1.11 \times 10^{-04}$	0

Table 7.1: Rounding errors when solving ill-conditioned system

Note the numerical value may actually vary with your machine.

Example 183. A notorious example is the fate of the Ariane rocket launched on June 4, 1996 (European Space Agency 1996). In the 37th second of flight, the inertial reference system attempted to convert a 64-bit floating-point number to a 16-bit number, but instead triggered an overflow error which was interpreted by the guidance system as flight data, causing the rocket to veer off course and be destroyed.[3]

Example 184. The Patriot missile defense system used during the Gulf War was also rendered ineffective due to roundoff error (Skeel 1992, U.S. GAO 1992). The system used an integer timing register which was incremented at intervals of 0.1s. However, the integers were converted to decimal numbers by multiplying by the binary approximation of 0.1,

$$0.00011001100110011001100_2 = \frac{209715}{2097152}.$$

As a result, after 100 hours, an error of

$$\left(\frac{1}{10} - \frac{209715}{2097152}\right) \times (3600 \times 100 \times 10) = \frac{5625}{16384} \approx 0.3433s$$

had accumulated. This discrepancy caused the Patriot system to continuously recycle itself instead of targeting properly. As a result, an Iraqi Scud missile could not be targeted and was allowed to detonate on a barracks, killing 28 people.

[3] See http://mathworld.wolfram.com/RoundoffError.html

7.2 Floating point numbers

Almost all processors use IEEE standard arithmetic, which is an arithmetic system based on a *floating* point representation of the real numbers combined with the binary system.

7.2.1 Binary system

Most computers deal with real numbers in binary systems, in contrast to the decimal system that humans prefer to use. The binary system uses 2 as the base in the same way that the decimal system uses 10. Consider the number 427.325. Then we have

$$427.375 = 4 \times 10^2 + 2 \times 10^1 + 7 \times 10^0 + 3 \times 10^{-1} + 7 \times 10^{-2} + 5 \times 10^{-3}.$$

The expression on the right contains powers of 10 and digits (here: 4, 2, 7, 3, 7, 5). If we accept the possibility of having an infinite number of digits to the right of the decimal point, then any real number can be expressed with a sign (+ or −) affixed to it. For example, $-\pi$ is

$$-\pi = -3.14159265358979323846264338\ldots$$

In the binary system, only the two digits 0 and 1 are used. A number in the binary system can be expressed in a similar manner. For example, we have

$$\begin{aligned} 427.375 = 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 \\ + 1 \times 2^{-2} + 1 \times 2^{-3}. \end{aligned} \quad (7.2)$$

The base 2 number

$$\begin{aligned} (1001.11101)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5} \end{aligned} \quad (7.3)$$

corresponds to the number 9.90625.

To compute the binary representation, we perform a series of division by 2 and reverse the order of the remainders to get the representation. For example:

- $(15)_{10} = (1111)_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
  - $15 = 2 * 7 + 1 \ \rightarrow$ remainder 1
  - $7 = 2 * 3 + 1 \ \rightarrow$ remainder 1
  - $3 = 2 * 1 + 1 \ \rightarrow$ remainder 1
  - $1 = 2 * 0 + 1 \ \rightarrow$ remainder 1
- $(11)_{10} = (1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
  - $11 = 2 * 5 + 1 \ \rightarrow$ remainder 1

- $5 = 2 * 2 + 1 \ \rightarrow$ remainder 1
- $2 = 2 * 1 + 0 \ \rightarrow$ remainder 0
- $1 = 2 * 0 + 1 \ \rightarrow$ remainder 1

- $(0.375)_{10} = 0.25 + 0.125 = 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = (0.011)_2$
  - $0.375 * 2 = 0.75 \ \rightarrow$ integral part 0
  - $0.75 * 2 = 1.5 \ \rightarrow$ integral part 1
  - $0.5 * 2 = (1.5 - 1.0) * 2 = 1.0 \ \rightarrow$ integral part 1

- $(19.625)_{10} = 16 + 2 + 1 + 0.5 + 0.125 = 1 \times 2^4 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} = (10011.101)_2$
  - $19 = 2 * 9 + 1 \ \rightarrow$ remainder 1
  - $9 = 2 * 4 + 1 \ \rightarrow$ remainder 1
  - $4 = 2 * 2 + 0 \ \rightarrow$ remainder 0
  - $2 = 2 * 1 + 0 \ \rightarrow$ remainder 0
  - $1 = 2 * 0 + 1 \ \rightarrow$ remainder 1
  - $0.625 * 2 = 1.25 \ \rightarrow$ integral part 1
  - $0.25 * 2 = (1.25 - 1.0) * 2 = 0.5 \ \rightarrow$ integral part 0
  - $0.5 * 2 = 1.0 \ \rightarrow$ integral part 1

In general, any integer $\beta > 1$ can be used as the base for a number system. Numbers represented in base β will contain digits $0, 1, 2, \ldots, \beta - 1$. The notation

$$(x)_\beta = (-1)^s \left(x_n x_{n-1} \ldots x_1 x_0 . x_{-1} x_{-2} \ldots x_{-m}\right)_\beta \quad \text{with } x_n \neq 0 \tag{7.4}$$

means

$$\begin{aligned}(x)_\beta &= (-1)^s \left(\sum_{k=-m}^{n} x_k \beta^k \right) \\ &= (-1)^s (x_n \times \beta^n + x_{n-1} \times \beta^{n-1} + \ldots + x_1 \times \beta + x_0 \times 1 \\ &\qquad + x_{-1} \times \frac{1}{\beta} + x_{-2} \times \frac{1}{\beta^2} + \ldots + x_{-m} \times \frac{1}{\beta^m}).\end{aligned} \tag{7.5}$$

Computers communicate with users in the decimal system but works internally in the binary system. Conversion procedures must be executed by the computer. We should be aware that there are two conversions involved — from and to decimal. Errors can be involved in each conversion.

Recall that computers represent only a subset of the real numbers. For example, a simple number like 1/10 can not be stored exactly on any binary machine. It requires an infinite binary expression

$$\frac{1}{10} = (0.0001\,1001\,1001\,1001\,\ldots)_2 .$$

So the conversion of 1/10 to binary will result in a roundoff error.

7.2.2 Floating point number

Almost all processors use IEEE standard arithmetic, which is an arithmetic system based on a *floating* point representation of the real numbers. In a floating point number system, the position of the decimal (or binary) point is stored separately from the digits. The gaps between adjacent represented numbers scale in proportion to the size of the numbers. This is different from a *fixed* point representation where the gaps are all the same.

In normalized scientific notation, any nonzero real number x can be represented as

$$x = \pm m \times 10^e$$

where the mantissa m is a number in the range $\frac{1}{10} \leq m < 1$ and the exponent e is an integer (positive, negative, or zero). When $x = 0$, then $e = 0$.

In the binary system, we have

$$x = \pm m \times 2^e$$

where $\frac{1}{2} \leq m < 1$ (if $x \neq 0$). Both m and e are base 2 numbers.

IEEE arithmetic includes two kinds of floating point numbers: single precision (32 bits long) and double precision (64 bits long). If s, e, and $m < 1$ are the 1-bit sign, 8-bit exponent, and 23-bit mantissa (or fraction) in the IEEE single precision format, then the number represented is

$$(-1)^s \times 2^{e-127} \times (1 + m).$$

On 8 bits, the exponent satisfy

$$0 < e < (11\,111\,111)_2 = 2^8 - 1 = 255.$$

The values $e = 0$ and $e = 255$ are reserved for special cases such as 0, $\pm\infty$, and `NaN`. With a 23-bit mantissa, the least significant bit in the mantissa represents units of 2^{-23} (or approximately 1.2×10^{-7}). The range of positive normalized numbers is from 2^{-126} (the underflow threshold) to $2^{127} \times (2 - 2^{-23}) \approx 2^{128}$ (the overflow threshold) or about 10^{-38} to 10^{38}.

If s, e, and $m < 1$ are the 1-bit sign, 11-bit exponent, and 52-bit mantissa (or fraction) in the IEEE double precision format, then the number represented is

$$(-1)^s \times 2^{e-1023} \times (1 + m).$$

On 11 bits, the exponent satisfy

$$0 < e < (11\,111\,111)_2 = 2^{11} - 1 = 2047.$$

The maximum relative representation error is $2^{-53} \approx 10^{-16}$. The range of positive normalized numbers is from 2^{-1022} (the underflow threshold) to $2^{1023} \times (2 - 2^{-52}) \approx 2^{1024}$ (the overflow threshold) or about 10^{-308} to 10^{308}.

7.2.3 Machine epsilon

The machine epsilon is traditionally the distance between 1 and the next larger floating point number, $1 + 2^{-52}$. It is also defined as the maximum relative representation error. In a relative sense, the machine epsilon $\epsilon_{\text{machine}}$ has the property

$$\forall x \in \mathbb{R}, \text{ there exists a floating number } x', \ |x - x'| \leq \epsilon_{\text{machine}}|x'|.$$

7.2.4 Special symbols Inf and NaN

The two error values are "infinity" (often denoted "Inf"), and "NaN" ("not a number"), which covers all other errors. Inf does not necessarily mean that the result is actually infinite. It simply means "too large to represent".

Both of these are encoded with the exponent field set to all 1s. (Recall that exponent fields of all 0s or all 1s are reserved for special meanings.) The mantissa field is set to something that can distinguish them—typically zero for Inf and nonzero for NaN. The sign bit is meaningful for Inf, that is, floating-point hardware distinguishes between + and −.

When a nonzero number is divided by zero (the divisor must be exactly zero), a "divide by zero" event occurs, and the result is set to infinity of the appropriate sign. In other cases in which the result's exponent is too large to represent, such as division of an extremely large number by an extremely small number, an "overflow" event occurs, also producing infinity of the appropriate sign. This is different from a divide by zero, though both produce a result of infinity, and the distinction is usually unimportant in practice.

Floating-point hardware is generally designed to handle operands of infinity in a reasonable way, such as

- (+Inf) + (+7) = (+Inf)
- (+Inf) × (−2) = (−Inf)
- (+Inf) × 0 = NaN — there is no meaningful thing to do

Remark 185. Wikipedia has the following interesting remark.

> *The advantage of floating-point representation over fixed-point (and integer) representation is that it can support a much wider range of values. For example, a fixed-point representation that has seven decimal digits with two decimal places, can represent the numbers 12345.67, 123.45, 1.23 and so on, whereas a floating-point representation with seven decimal digits could in addition represent 1.234567, 123456.7, 0.00001234567, 1234567000000000, and so on. The floating-point format needs slightly more storage (to encode the position of the radix point), so when stored in the same space, floating-point numbers achieve their greater range at the expense of precision.*

7.3 Floating point operations

7.3.1 Arithmetic

On a computer, all mathematical computations are reduced to certain elementary arithmetic operations. The classical set is $+$, $-$, $\times$, $\div$. Mathematically, these symbols represent operations in $\mathbb{R}$. The corresponding computer operations involve roundoff errors.

Denote $\mathrm{fl}(x)$ the floating point number closest to x. Then we have

$$\mathrm{fl}(x) = x(1+\varepsilon) \quad \text{with} \quad |\varepsilon| \leq \epsilon_{\text{machine}}.$$

Note that we have $\mathrm{fl}(\mathrm{fl}(x)) = \mathrm{fl}(x)$. In computer arithmetic, the four basic operations applied to floating point numbers are exact up to a relative error of size at most $\epsilon_{\text{machine}}$. For example, for the substraction, we have

$$\mathrm{fl}(\mathrm{fl}(x) \ominus \mathrm{fl}(y)) = (\mathrm{fl}(x) - \mathrm{fl}(y))(1+\epsilon) \quad \text{with} \quad |\epsilon| \leq \epsilon_{\text{machine}} \tag{7.6}$$

and

$$\mathrm{fl}(\mathrm{fl}(x) \ominus \mathrm{fl}(y)) = (x(1+\varepsilon_1) - y(1+\varepsilon_2))(1+\epsilon) \quad \text{with} \quad |\epsilon| \leq \epsilon_{\text{machine}}. \tag{7.7}$$

7.3.2 Operation counts

The work required to solve a problem on a computer is often measured in terms of the number of floating point operations or flops needed to do the calculation. A floating point operation is multiplication, division, addition, or subtraction on floating point numbers.

Example 186. Computing the inner product of two vectors in $\mathbb{R}^n$,

$$\mathbf{x}^T\mathbf{y} = x_1y_1 + x_2y_2 + \ldots + x_ny_n,$$

requires n multiplications and $n-1$ additions for a total of $2n-1$ flops.

Example 187. Computing $\mathbf{y} = \mathbf{A}\mathbf{x}$ where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{x} \in \mathbb{R}^n$. The i-th element of $\mathbf{y}$ is the inner product of the i-th row of $\mathbf{A}$ with $\mathbf{x}$ and requires $2n-1$ flops. There are m rows. So we need to compute y_i for $i = 1, 2, \ldots, m$. The total work is $2mn - m$ flops. When the matrix $\mathbf{A}$ is square ($m = n$), the count becomes $2n^2 - n$. For large value of n, the first term dominates and we say the work required is $\mathcal{O}(n^2)$ as n goes to $+\infty$.

If the work required by some algorithm for a system of size n has the flop count

$$W(n) = a_0 + a_1n + a_2n^2 + \ldots + a_kn^k$$

(where a_j is a constant), then the last term dominates for large values of n. We say

$$W(n) = \mathcal{O}(n^k) \quad \text{as } n \longrightarrow +\infty.$$

More generally, if $W(n)$ is "Big oh" of n^k , then the ratio $W(n)/n^k$ remains bounded as $n \longrightarrow +\infty$:

$$W(n) = \mathcal{O}(n^k) \quad \Leftrightarrow \quad 0 \leq W(n) \leq Cn^k \quad \text{for large values of } n$$

(where C is a constant independent on n).

Remark. When $W(n) = \mathcal{O}(n^k)$, then doubling the size of the problem (from n to $2n$) will increase the work by a factor 2^k.

Example 188. Consider $\mathbf{A} \in \mathbb{R}^{m\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times n}$. Computing the matrix-matrix product $\mathbf{C} = \mathbf{AB}$ will require the computations of mn entries c_{ij}. Each entry c_{ij} is the inner product of two vectors with r components, which require $2r - 1$ flops. So the total work becomes $2mnr - mn$ flops. When $m = n = r$, the matrix-matrix product requires $\mathcal{O}(n^3)$ flops.

You may know how "fast" your computer processor is, e.g. 2 GHz (2 gigahertz, or 2 billion cycles per second). This tells something about how many instructions the processor can execute per second. When using numerical methods we generally care most about how many *floating point operations per second* or flops it can perform, where a floating point operation means adding two numbers together, or other basic arithmetic operation. This can vary quite a bit depending on exactly what is being done, but generally has the same order of magnitude as the cycle speed (a 2 GHz processor can do roughly 1 billion arithmetic operations per second). We use the terms megaflops, gigaflops, teraflops, etc. to talk about the speed of computers.

Currently the "fastest" computer in the world is the Roadrunner computer at Los Alamos National Laboratory, which can perform certain benchmark linear algebra problems at more than one petaflops (10^{15} floating point operations per second). See `http://www.top500.org` for a list of the top 500 machines in the world and information about each. It should be noted, however, that these fast machines are based on processors that are not all that fast, but are composed of thousands of processors working together. To get this fast performance it is necessary to develop algorithms that can use all these processors together to solve a single problem. This is not easy, and one of the main challenges in numerical methods for the future is to develop better ways of using parallel processors.

You should know the common prefixes (the bigger ones will become more common in the future!):

- kilo: 10^3
- mega: 10^6
- giga: 10^9
- tera: 10^{12}
- peta: 10^{15}
- exa: 10^{18}

7.4 Stability

Since most problems are continuous while digital computers are discrete, numerical algorithms can generally not provide the exact solution. The notion of stability is the standard way of characterizing what is possible. It corresponds to the numerical analysts' idea of what it means to get the "right answer", even if it is not exact.

Consider a mathematical problem as a function f between two linear spaces ($f : X \to Y$) and an algorithm as another map $\tilde{f}$ between the same two spaces.

Definition 189. An algorithm $\tilde{f}$ for a problem f is accurate if we have

$$\forall x \in X, \quad 0 \leq \frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq C\epsilon_{\text{machine}} = \mathcal{O}(\epsilon_{\text{machine}}) \tag{7.8}$$

where C is a constant independent of x.

Accuracy is a strong statement that is difficult to attain when the problem f is ill-conditioned. The next two definitions are less restrictive.

Definition 190. An algorithm $\tilde{f}$ for a problem f is stable if we have

$$\forall x \in X,\ \exists\ \tilde{x} \in X \text{ such that } \frac{\|\tilde{x} - x\|}{\|x\|} = \mathcal{O}(\epsilon_{\text{machine}})$$
$$\text{and} \quad \frac{\|\tilde{f}(x) - f(\tilde{x})\|}{\|f(\tilde{x})\|} = \mathcal{O}(\epsilon_{\text{machine}}) \tag{7.9}$$

A stable algorithm gives nearly the right answer to nearly the right question.

Many algorithms in numerical linear algebra satisfy a condition that is both stronger and simpler than stability.

Definition 191. An algorithm $\tilde{f}$ for a problem f is backward stable if we have

$$\forall x \in X,\ \exists\ \tilde{x} \in X \text{ such that } \frac{\|\tilde{x} - x\|}{\|x\|} = \mathcal{O}(\epsilon_{\text{machine}}) \quad \text{and} \quad \tilde{f}(x) = f(\tilde{x}). \tag{7.10}$$

A backward stable algorithm gives exactly the right answer to nearly the right question.

In this course, we will not analyze algorithms from the point of view of stability. Instead, we will discuss some examples. This is an important research field. The book of Trefethen and Bau[4] gives a nice introduction.

7.5 Example

Consider the classical Gram-Schmidt algorithm used in the QR factorization (see Algorithm 3)

[4] L. N. Trefethen and D. Bau, *Numerical linear algebra*, SIAM, Philadelphia, 1997.

Algorithm 3 Classical Gram-Schmidt

```
for j = 1 to n do
  v = a_j
  for i = 1 to j − 1 do
    r_ij = q_i^T a_j
    v = v − r_ij q_i
  end for
  r_jj = ‖v‖_2
  q_j = v/r_jj
end for
```

First, we compute the number of operations. The computation of r_{ij} is an inner product between two vectors in $\mathbb{R}^m$, which costs $2m-1$ flops. The update of $\mathbf{v}$ requires m multiplications and m substractions. The total work is a single inner iteration is $4m-1$ flops. All together, the number of flops required by the algorithm is

$$\sum_{j=1}^{n}\left[\left(\sum_{i=1}^{j-1}4m-1\right)+2m-1+1\right]=\sum_{j=1}^{n}[(4m-1)(j-1)+2m]$$
$$=(4m-1)\left(\sum_{j=1}^{n}j\right)-(2m-1)n \quad (7.11)$$

We recall the useful formula

$$\sum_{k=1}^{n}k=\frac{k(k+1)}{2}.$$

So the total number of flops for the classical Gram-Schmidt algorithm is

$$(4m-1)\frac{n(n+1)}{2}-(2m-1)n\approx 2mn^2=\mathcal{O}(mn^2).$$

The classical Gram-Schmidt algorithm is numerically unstable because of rounding errors on a computer. To illustrate the unstability, consider the Hilbert matrix

$$\mathbf{H}_4=\begin{bmatrix}1 & 1/2 & 1/3 & 1/4\\ 1/2 & 1/3 & 1/4 & 1/5\\ 1/3 & 1/4 & 1/5 & 1/6\\ 1/4 & 1/5 & 1/6 & 1/7\end{bmatrix}.$$

A MATLAB implementation of Algorithm 3 generates a matrix $\mathbf{Q}$ such that

$$\|\mathbf{Q}^T\mathbf{Q}-\mathbf{I}\|_1\approx 3.6\times 10^{-11} \quad (7.12)$$

Fortunately, there are simple modifications that improve matters. For example, we describe, in Algorithm 4, the modified Gram-Schmidt which requires

Algorithm 4 Modified Gram-Schmidt

$\mathbf{V} = \mathbf{A}$
for $i = 1$ to n **do**
  $r_{ii} = \|\mathbf{v}_i\|_2$
  $\mathbf{q}_i = \mathbf{v}_i / r_{ii}$
  for $j = i + 1$ to n **do**
    $r_{ij} = \mathbf{q}_i^T \mathbf{v}_j$
    $\mathbf{v}_j = \mathbf{v}_j - r_{ij}\mathbf{q}_i$
  end for
end for

the same amount of flops than the classical Gram-Schmidt algorithm. For the Hilbert matrix $\mathbf{H}_4$, the modified Gram-Schmidt algorithm will compute a matrix $\mathbf{Q}$ such that

$$\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}\|_1 \approx 4.0 \times 10^{-13} \tag{7.13}$$

which is better than (7.12).

Another possibility is to apply the classical Gram-Schmidt algorithm twice, as described in Algorithm 5. The total number of flops is asymptotically equal to $4mn^2$. For the Hilbert matrix $\mathbf{H}_4$, applying twice the classical Gram-Schmidt

Algorithm 5 Two steps of classical Gram-Schmidt

$\mathbf{R} = \mathbf{I}$
for $j = 1$ to n **do**
  $\mathbf{v} = \mathbf{a}_j$
  for $i = 1$ to $j - 1$ **do**
    $\tilde{r}_{ij} = \mathbf{q}_i^T \mathbf{a}_j$
    $\mathbf{v} = \mathbf{v} - \tilde{r}_{ij}\mathbf{q}_i$
  end for
  $\tilde{r}_{jj} = \|\mathbf{v}\|_2$
  $\mathbf{q}_j = \mathbf{v}/\tilde{r}_{jj}$
end for
$\mathbf{R} = \tilde{\mathbf{R}}\mathbf{R}$
for $j = 1$ to n **do**
  $\mathbf{v} = \mathbf{q}_j$
  for $i = 1$ to $j - 1$ **do**
    $\tilde{r}_{ij} = \mathbf{q}_i^T \mathbf{q}_j$
    $\mathbf{v} = \mathbf{v} - \tilde{r}_{ij}\mathbf{q}_i$
  end for
  $\tilde{r}_{jj} = \|\mathbf{v}\|_2$
  $\mathbf{q}_j = \mathbf{v}/\tilde{r}_{jj}$
end for
$\mathbf{R} = \tilde{\mathbf{R}}\mathbf{R}$

algorithm will compute a matrix $\mathbf{Q}$ such that

$$\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}\|_1 \approx 3.6 \times 10^{-16}. \tag{7.14}$$

Remark. The MATLAB function `qr` uses the Householder triangularization, which is an advanced algorithm (see Trefethen and Bau[5]). It requires, asymptotically, $2mn^2 - \frac{2}{3}n^3$ flops.

We can compare these four algorithms for a series of Hilbert matrices. Recall that the Hilbert matrix gets quickly ill-conditioned (see Table 7.2).

	Condition Number
$\mathbf{H}_4$	$2.8 \times 10^{+04}$
$\mathbf{H}_5$	$9.4 \times 10^{+05}$
$\mathbf{H}_6$	$2.9 \times 10^{+07}$
$\mathbf{H}_7$	$9.8 \times 10^{+08}$
$\mathbf{H}_8$	$3.4 \times 10^{+10}$

Table 7.2: Condition number for Hilbert matrices using the 1-norm

Table 7.3 describes the resulting error when the dimension of the Hilbert matrix is increased.

	Classical G-S	Modified G-S	Two-Step G-S	MATLAB `qr`
$\mathbf{H}_4$	3.6×10^{-11}	4.0×10^{-13}	3.6×10^{-16}	7.5×10^{-16}
$\mathbf{H}_5$	1.1×10^{-08}	9.0×10^{-12}	3.2×10^{-16}	1.3×10^{-15}
$\mathbf{H}_6$	1.7×10^{-04}	6.9×10^{-10}	2.5×10^{-16}	9.2×10^{-16}
$\mathbf{H}_7$	5.2×10^{-02}	5.6×10^{-09}	4.9×10^{-16}	1.1×10^{-15}
$\mathbf{H}_8$	1.1×10^{-00}	3.4×10^{-07}	3.9×10^{-13}	1.4×10^{-15}

Table 7.3: Error $\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}\|_1$ with different algorithms for Hilbert matrices

The results highlight that the classical Gram-Schmidt algorithm is unstable. The modified Gram-Schmidt algorithm computes an approximate orthogonal matrix with an error proportional to the condition number. When applying twice the classical Gram-Schmidt algorithm, the error is close to machine precision. It increases only for $\mathbf{H}_8$ but a third step of Gram-Schmidt would fix the problem. Note that the MATLAB function returns an orthogonal matrix up to machine precision.

7.6 Useful commands in MATLAB

Here are a few commands in MATLAB useful for this section.

- `eps` contains the machine epsilon.

[5] L. N. Trefethen and D. Bau, *Numerical linear algebra*, SIAM, Philadelphia, 1997.

- `bin2dec` converts binary string to decimal integer.
- `dec2bin` converts decimal integer to a binary string.
- `format` sets the output format. For example, `format long e` selects the floating point format with 15 digits for double and 7 digits for single.
- `intmax` returns the largest positive integer value.
- `intmin` returns the smallest positive integer value.
- `realmax` returns the largest positive floating point number.
- `double` converts to double precision.
- `single` converts to single precision.

7.7 Exercises

Exercise 192. Convert the following numbers to binary (Detail your derivation!)

- $31 = (31)_{10}$
- $32 = (32)_{10}$
- $6.125 = (6.125)_{10}$
- $18.3125 = (18.3125)_{10}$
- $0.1 = (0.1)_{10}$

8 Linear systems of equations

A linear system of m equations in n unknowns $x_1, x_2, \ldots, x_n$ has the general form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned} \tag{8.1}$$

where the a_{ij} and b_j are some given real numbers. This can be written as

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{8.2}$$

where $\mathbf{A}$ is an $m \times n$ matrix and $b \in \mathbb{R}^m$. The problem is to find a vector $\mathbf{x} \in \mathbb{R}^n$ so that (8.2) holds.

We have seen that we can view matrix-vector multiplication as a linear combination of the columns of the matrix $\mathbf{A}$. So another way to write this system is

$$\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} x_1 + \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} x_2 + \cdots + \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} x_n = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}. \tag{8.3}$$

In other words, we are trying to find the weights $x_1, x_2, \ldots, x_n$ so that the vector $\mathbf{b}$ can be expressed as a linear combination of the columns of $\mathbf{A}$. The objective of this Section is to describe general algorithms to solve such linear systems.

8.1 Counting the number of solutions

When solving linear system is viewed as trying to find the weights of a linear combination (like (8.3)), we see that a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ has at least one solution if (and only if) the vector $\mathbf{b}$ lies in the subspace of $\mathbb{R}^m$ spanned by the columns of $\mathbf{A}$. We recall that the range of $\mathbf{A}$, $\mathcal{R}(\mathbf{A})$, is exactly the subspace spanned by the columns of $\mathbf{A}$.

A linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ has at least one solution if $\mathbf{b} \in \mathcal{R}(\mathbf{A})$. Might it have more than one solution? Suppose the system has two distinct solutions, *i.e.* $\mathbf{A}\mathbf{x} = \mathbf{b}$ and also $\mathbf{A}\mathbf{y} = \mathbf{b}$ where $\mathbf{x} \neq \mathbf{y}$. Let $\mathbf{z}$ be the difference between $\mathbf{x}$ and $\mathbf{y}$: $\mathbf{z} = \mathbf{x} - \mathbf{y} \in \mathbb{R}^n$. Then we have

$$\mathbf{A}\mathbf{z} = \mathbf{A}(\mathbf{x} - \mathbf{y}) = \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y} = \mathbf{b} - \mathbf{b} = \mathbf{0}.$$

We have found a nontrivial linear combination (since $\mathbf{z} \neq \mathbf{0}$) of the columns of $\mathbf{A}$ that gives the zero vector. Hence the columns of $\mathbf{A}$ are linearly dependent. From this, we conclude:

- If a linear system has more than one solution then $\text{rank}(\mathbf{A}) < n$.

Conversely, suppose $\text{rank}(\mathbf{A}) < n$. Then the columns of $\mathbf{A}$ are linearly dependent and the null space of $\mathbf{A}$, $\mathcal{N}(\mathbf{A})$, is non trivial. So there must be a vector $\mathbf{z} \neq \mathbf{0}$ for which $\mathbf{Az} = \mathbf{0}$. Now suppose $\mathbf{b} \in \mathcal{R}(\mathbf{A})$ and we have one solution $\mathbf{x}$ to $\mathbf{Ax} = \mathbf{b}$. Then $\mathbf{x} + \mathbf{z} \neq \mathbf{x}$ is also a solution, since

$$\mathbf{A}(\mathbf{x} + \mathbf{z}) = \mathbf{Ax} + \mathbf{Az} = \mathbf{b} + \mathbf{0} = \mathbf{b}.$$

Also note that if $\mathbf{Az} = \mathbf{0}$ then $\mathbf{A}(\alpha\mathbf{z}) = \mathbf{0}$ for any scalar $\alpha \in \mathbb{R}$, so in this case there are *infinitely* many different solutions to the linear system.

We have proved the following theorem:

Theorem 193. *Consider the linear system* $\mathbf{Ax} = \mathbf{b}$ *with* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{b} \in \mathbb{R}^m$*. Then:*

- *If* $\mathbf{b} \notin \mathcal{R}(\mathbf{A})$*, then the system has no solution.*
- *If* $\mathbf{b} \in \mathcal{R}(\mathbf{A})$ *and* $\text{rank}(\mathbf{A}) = n$*, then this system has a unique solution* $\mathbf{x}$*.*
- *If* $\mathbf{b} \in \mathcal{R}(\mathbf{A})$ *and* $\text{rank}(\mathbf{A}) < n$*, then this system has infinitely many solutions.*

A special case that often arises in practice is a system of n equations in n unknowns, in which case $m = n$ in our previous discussion. All the statements made above still apply in this case, but now we see that there are basically two cases:

- If $\text{rank}(\mathbf{A}) = n$ then $\mathcal{R}(\mathbf{A}) = \mathbb{R}^n$ and $\mathcal{N}(\mathbf{A}) = \{\mathbf{0}\}$. In this case, we say the matrix $\mathbf{A}$ is nonsingular or invertible. The system $\mathbf{Ax} = \mathbf{b}$ has *exactly* one solution for *any* right hand side $\mathbf{b} \in \mathbb{R}^n$.
- If $\text{rank}(\mathbf{A}) < n$ then the range of $\mathbf{A}$ is a proper subspace of $\mathbb{R}^n$ (not the full space) and so $\mathbf{Ax} = \mathbf{b}$ may or may not have solutions, depending on whether $\mathbf{b} \in \mathcal{R}(\mathbf{A})$ or not. If it does have solutions then it has *infinitely* many solutions, since the null space of $\mathbf{A}$ has dimension $n - \text{rank}(\mathbf{A}) \geq 1$. The matrix $\mathbf{A}$ is singular or noninvertible.

Example 194. The linear system

$$\begin{bmatrix} 3 & 0 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 14 \end{bmatrix}$$

has exactly one solution $\begin{bmatrix} 1 \\ 3 \end{bmatrix}$. We can change the right hand side of this system to any other vector and the system will still have exactly one solution.

Example 195. The linear system

$$\begin{bmatrix} 2 & 6 \\ -5 & 3 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 1 \\ 5 \end{bmatrix}$$

has exactly one solution $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$. The columns of $\mathbf{A}$ are linearly independent (since the second is not a scalar multiple of the first). The matrix is of full rank and its null space is equal to $\{\mathbf{0}\}$. On the other hand, the system

$$\begin{bmatrix} 2 & 6 \\ -5 & 3 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 5 \end{bmatrix}$$

has no solution.

Example 196. The linear system

$$\begin{bmatrix} 3 & 0 & -1 \\ 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 14 \end{bmatrix}$$

has infinitely many solutions,

$$\mathbf{x} = \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ -2 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 8 \\ -3 \end{bmatrix}, \ldots$$

The difference between any two of these solutions lies in the null space of $\mathbf{A}$, which is

$$\mathcal{N}(\mathbf{A}) = \text{span}\left(\begin{bmatrix} 1 \\ -5 \\ 3 \end{bmatrix} \right).$$

We also know the null space should have dimension 1 since the rank of $\mathbf{A}$ is 2 and so $dim(\mathcal{N}(\mathbf{A})) = 3 - 2 = 1$.

In the following sections, we will discuss different algorithms to solve a linear system $\mathbf{Ax} = \mathbf{b}$.

8.2 Exercises

9 LU Factorization

In this section, we study the solution of linear systems $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is an invertible matrix.

9.1 Easy-to-solve systems

We begin by looking for special types of systems that can be easily solved.

Suppose that the $n \times n$ matrix $\mathbf{A}$ has a diagonal structure. This means that all the nonzero elements of $\mathbf{A}$ are on the diagonal of $\mathbf{A}$. A linear system $\mathbf{Ax} = \mathbf{b}$ is

$$\begin{bmatrix} a_{11} & 0 & \cdots & & 0 \\ 0 & a_{22} & & & 0 \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & & \cdots & 0 & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{bmatrix}.$$

The system collapses to n simple equations and the solution is

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1/a_{11} \\ b_2/a_{22} \\ \vdots \\ \vdots \\ b_n/a_{nn} \end{bmatrix}.$$

The computation requires n floating operations. If $a_{ii} = 0$ and $b_i = 0$ for some index i, then x_i can be any real number. If $a_{ii} = 0$ and $b_i \neq 0$ for some index i, then no solution exists. Note that for a diagonal matrix, its determinant is

$$\det \mathbf{A} = a_{11} \times a_{22} \times \cdots \times a_{nn} = \prod_{i=1}^{n} a_{ii}.$$

Example 197. The linear system

$$\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 14 \end{bmatrix}$$

is lower triangular. So we obtain $x_1 = 3/3 = 1$ and $2x_2 = 14$. So the exact solution is $\begin{bmatrix} 1 \\ 7 \end{bmatrix}$.

Suppose that the $n \times n$ matrix $\mathbf{A}$ has a lower triangular structure. This means that all the nonzero elements of $\mathbf{A}$ are below the main diagonal of $\mathbf{A}$. A

linear system $\mathbf{Ax} = \mathbf{b}$ is

$$\begin{bmatrix} a_{11} & 0 & \cdots & & 0 \\ a_{21} & a_{22} & & & 0 \\ \vdots & & \ddots & & \vdots \\ & & & \ddots & 0 \\ a_{n1} & a_{n2} & \cdots & & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{bmatrix}.$$

Assume that $a_{ii} \neq 0$ for all indices i. Then we obtain x_1 from the first equation. With the known value of x_1 substituted in the second equation, we can solve for x_2. We proceed in the same way obtaining $x_1, x_2, \cdots, x_n$ one at a time and in this order. A formal algorithm for this case is called **forward substitution**. The total number of operations is $n(n+2) = \mathcal{O}(n^2)$. The solution is

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1/a_{11} \\ (b_2 - a_{21}x_1)/a_{22} \\ \vdots \\ \vdots \\ (b_n - \sum_{j=1}^{n-1} a_{nj}x_j)/a_{nn} \end{bmatrix}.$$

If $a_{ii} = 0$ for some index i, then the matrix is singular. Its determinant is

$$\det \mathbf{A} = a_{11} \times a_{22} \times \cdots \times a_{nn} = \prod_{i=1}^{n} a_{ii}.$$

If $a_{ii} = 0$ and $b_i - \sum_{j=1}^{i-1} a_{ij}x_j = 0$ for some index i, then x_i can be any real number. If $a_{ii} = 0$ and $b_i \neq \sum_{j=1}^{i-1} a_{ij}x_j$ for some index i, then no solution exists.

Example 198. The linear system

$$\begin{bmatrix} 3 & 0 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 14 \end{bmatrix}$$

is lower triangular. So we obtain $x_1 = 3/3 = 1$ and $4x_2 = 14 - 2 \times 1 = 12$. So the exact solution is $\begin{bmatrix} 1 \\ 3 \end{bmatrix}$.

Suppose that the $n \times n$ matrix $\mathbf{A}$ has an upper triangular structure. This means that all the nonzero elements of $\mathbf{A}$ are below the main diagonal of $\mathbf{A}$. A linear system $\mathbf{Ax} = \mathbf{b}$ is

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & & a_{1n} \\ 0 & a_{22} & & & a_{2n} \\ \vdots & 0 & \ddots & & \vdots \\ & & & \ddots & \\ 0 & 0 & \cdots & 0 & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{bmatrix}.$$

Assume that $a_{ii} \neq 0$ for all indices i. Then we obtain x_n from the last equation. With the known value of x_n substituted in the next-to-last equation, we can solve for x_{n-1}. We proceed in the same way obtaining $x_n, x_{n-1}, \cdots, x_1$ one at a time and in this order. A formal algorithm for this case is called **backward substitution**. The total number of operations is still $\mathcal{O}(n^2)$. If $a_{ii} = 0$ for some index i, then the matrix is singular. Its determinant is

$$\det \mathbf{A} = a_{11} \times a_{22} \times \cdots \times a_{nn} = \prod_{i=1}^{n} a_{ii}.$$

If $a_{ii} = 0$ and $b_i - \sum_{j=1}^{i-1} a_{ij} x_j = 0$ for some index i, then x_i can be any real number. If $a_{ii} = 0$ and $b_i \neq \sum_{j=1}^{i-1} a_{ij} x_j$ for some index i, then no solution exists.

Another simple type of system than can be easily solved using the same ideas, is a system obtained by permuting the equations in a triangular system. Consider the system

$$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & a_{23} \\ a_{31} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

If we simply reorder these equations, we can get a lower triangular system:

$$\begin{bmatrix} a_{31} & 0 & 0 \\ a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_3 \\ b_1 \\ b_2 \end{bmatrix}.$$

Then we can solve the re-ordered system in the same manner as previously. But the remaining question is how to determine beforehand the permutation. We will discuss a general approach in the next subsection.

9.2 LU factorization

The LU factorization is a simple way to solve a linear system by hand and also the standard way for solving linear systems on computers. The LU factorization transforms a full linear system into an upper triangular one by applying simple linear transformations.

Assume that $\mathbf{A} \in \mathbb{R}^{n \times n}$. A lower triangular matrix with diagonal entries equal to 1 can usually be found such that $\mathbf{L}^{-1}\mathbf{A} = \mathbf{U}$, where $\mathbf{U}$ is an upper triangular matrix. Thus we obtain an LU factorization of $\mathbf{A}$, *i.e.* $\mathbf{A} = \mathbf{LU}$. Before giving the general algorithm, we discuss some examples.

Example 199. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}.$$

The first step of the LU factorization looks like

$$\mathbf{L}_1\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 3 & 5 & 5 \\ 0 & 4 & 6 & 8 \end{bmatrix}.$$

We have subtracted twice the first row from the second, four times the first row from the third, and three times the first row from the fourth. The second step is

$$\mathbf{L}_2\mathbf{L}_1\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 3 & 5 & 5 \\ 0 & 4 & 6 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix}.$$

Here we have subtracted three times the second row from the third and four times the second row from the fourth. The final step is

$$\mathbf{L}_3\mathbf{L}_2\mathbf{L}_1\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix} = \mathbf{U}.$$

Now we need to compute the matrix $(\mathbf{L}_3\mathbf{L}_2\mathbf{L}_1)^{-1} = \mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\mathbf{L}_3^{-1}$. It turns out that the inverse of $\mathbf{L}_1$ is a simple modification of the matrix $\mathbf{L}_1$ where the terms below the diagonal have been negated,

$$\mathbf{L}_1^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{bmatrix}.$$

The inverses of $\mathbf{L}_2$ and $\mathbf{L}_3$ are obtained in a similar fashion. Note that this formula for the inverse is due to the special structure of the matrices $\mathbf{L}_i$ (that are equal to the identity matrix plus an outer product with only one nonzero column). The special structure of these matrices result also in simple matrix-matrix multiplications, *i.e.*

$$\mathbf{L}_1^{-1}\mathbf{L}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & 4 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 3 & 1 & 0 \\ 3 & 4 & 0 & 1 \end{bmatrix},$$

namely, we simply insert the nonzero subdiagonal entries in the appropriate places. Finally, we get

$$\mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\mathbf{L}_3^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 3 & 1 & 0 \\ 3 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 3 & 1 & 0 \\ 3 & 4 & 1 & 1 \end{bmatrix}$$

and

$$\mathbf{A} = \mathbf{LU} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 3 & 1 & 0 \\ 3 & 4 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}.$$

Example 200. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 10 \end{bmatrix}.$$

The LU factorization performs first

$$\mathbf{L}_1\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 10 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{bmatrix}.$$

Then the next step gives

$$\mathbf{L}_2\mathbf{L}_1\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{bmatrix}.$$

Finally, we write

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{bmatrix}.$$

Before describing the LU factorization, we recall that linear combinations of matrix rows or columns and permutations of rows and columns can be obtained by matrix-matrix multiplication. For example, the product of matrices

$$\begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4 \\ 5+2a & 6+3a & 7+4a \\ 8+2b & 9+3b & 10+4b \end{bmatrix} \tag{9.1}$$

corresponds to adding a times the first row to the second row and adding b times the first row to the third row. The product

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & c & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8+5c & 9+6c & 10+7c \end{bmatrix} \tag{9.2}$$

corresponds to adding c times the second row to the third row. The product

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} = \begin{bmatrix} 2a & 3a & 4a \\ 5b & 6b & 7b \\ 8c & 9c & 10c \end{bmatrix} \tag{9.3}$$

corresponds to row scalings. The column operations are easily obtained by transpose. For example, the product of matrices

$$\begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} \begin{bmatrix} 1 & \alpha & \beta \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3+2\alpha & 4+2\beta \\ 5 & 6+5\alpha & 7+5\beta \\ 8 & 9+8\alpha & 10+8\beta \end{bmatrix} \tag{9.4}$$

corresponds to adding α times the first column to the second column and adding β times the first column to the third column. The product

$$\begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \gamma \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4+3\gamma \\ 5 & 6 & 7+6\gamma \\ 8 & 9 & 10+9\gamma \end{bmatrix} \tag{9.5}$$

corresponds to adding γ times the second column to the third column. The product

$$\begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{bmatrix} = \begin{bmatrix} 2\alpha & 3\beta & 4\gamma \\ 5\alpha & 6\beta & 7\gamma \\ 8\alpha & 9\beta & 10\gamma \end{bmatrix} \tag{9.6}$$

corresponds to column scaling.

The general LU factorization goes as follows. Consider a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$. If $a_{11} \neq 0$, then we write

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ -a_{21}/a_{11} & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ -a_{n1}/a_{11} & 0 & & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & \tilde{a}_{22} & \cdots & \tilde{a}_{2n} \\ \vdots & & & \vdots \\ 0 & \tilde{a}_{n2} & & \tilde{a}_{nn} \end{bmatrix}.$$

We have eliminated all the entries below the diagonal in the first column. If $\tilde{a}_{22} \neq 0$, then we can repeat the process and eliminate all the entries below the diagonal in the second column. Suppose that we have build $p-1$ lower triangular matrices with diagonal entries equal to 1 such that the matrix $\mathbf{L}_{p-1}\ldots\mathbf{L}_2\mathbf{L}_1\mathbf{A}$ has the first $p-1$ columns with zero entries under the main diagonal, *i.e.*

$$\mathbf{L}_{p-1}\ldots\mathbf{L}_2\mathbf{L}_1\mathbf{A} = \begin{bmatrix} a_{11} & & & \cdots & & a_{1n} \\ 0 & \ddots & & & & \\ \vdots & \ddots & \ddots & & & \\ \vdots & & 0 & \hat{a}_{pp} & \ldots & \hat{a}_{pn} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \ldots & 0 & \hat{a}_{np} & \ldots & \hat{a}_{nn} \end{bmatrix}.$$

Then, if $\hat{a}_{pp} \neq 0$, we have

$$\begin{bmatrix} 1 & & & \cdots & & & & 0 \\ 0 & \ddots & & & & & & \\ \vdots & \ddots & \ddots & & & & & \\ \vdots & & 0 & 1 & & & & \\ \vdots & & \vdots & -\frac{\hat{a}_{p+1,p}}{\hat{a}_{pp}} & \ddots & & & \\ \vdots & & \vdots & \vdots & 0 & \ddots & & \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \ddots & \\ 0 & \dots & 0 & -\frac{\hat{a}_{n,p}}{\hat{a}_{pp}} & 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & & & \cdots & & a_{1n} \\ 0 & \ddots & & & & \\ \vdots & \ddots & \ddots & & & \\ \vdots & & 0 & \hat{a}_{pp} & \dots & \hat{a}_{pn} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & \hat{a}_{np} & \dots & \hat{a}_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & & & \cdots & & & a_{1n} \\ 0 & \ddots & & & & & \\ \vdots & \ddots & \ddots & & & & \\ \vdots & & 0 & \hat{a}_{pp} & & & \\ \vdots & & \vdots & 0 & \check{a}_{p+1,p+1} & \dots & \check{a}_{p+1,p+1} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & \check{a}_{nn} & \dots & \check{a}_{nn} \end{bmatrix}$$

By building $n-1$ lower triangular matrices with diagonal entries equal to 1, the resulting matrix is upper triangular.

When the lower triangular matrix $\mathbf{L}_p$ is

$$\mathbf{L}_p = \begin{bmatrix} 1 & & & \cdots & & & & 0 \\ 0 & \ddots & & & & & & \\ \vdots & \ddots & \ddots & & & & & \\ \vdots & & 0 & 1 & & & & \\ \vdots & & \vdots & -\frac{\hat{a}_{p+1,p}}{\hat{a}_{pp}} & \ddots & & & \\ \vdots & & \vdots & \vdots & 0 & \ddots & & \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \ddots & \\ 0 & \dots & 0 & -\frac{\hat{a}_{n,p}}{\hat{a}_{pp}} & 0 & \dots & 0 & 1 \end{bmatrix},$$

its inverse is

$$\mathbf{L}_p^{-1} = \begin{bmatrix} 1 & & & \cdots & & & & 0 \\ 0 & \ddots & & & & & & \\ \vdots & \ddots & \ddots & & & & & \\ \vdots & & 0 & 1 & & & & \\ \vdots & & \vdots & \frac{\hat{a}_{p+1,p}}{\hat{a}_{pp}} & \ddots & & & \\ \vdots & & \vdots & \vdots & 0 & \ddots & & \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \ddots & \\ 0 & \dots & 0 & \frac{\hat{a}_{n,p}}{\hat{a}_{pp}} & 0 & \dots & 0 & 1 \end{bmatrix}.$$

The product $\mathbf{L}_p^{-1}\mathbf{L}_{p+1}^{-1}$ is just the unit lower triangular matrix with the entries of both $\mathbf{L}_p^{-1}$ and $\mathbf{L}_{p+1}^{-1}$ inserted in their usual places below the diagonal. When we take the product of all of these matrices to form $\mathbf{L}$, we have the same convenient property everywhere below the diagonal

$$\mathbf{L} = \mathbf{L}_1^{-1} \dots \mathbf{L}_{n-1}^{-1} = \begin{bmatrix} 1 & 0 & & \cdots & 0 \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & & \ddots & \ddots & 0 \\ l_{m1} & l_{m2} & \dots & l_{m,m-1} & 1 \end{bmatrix}.$$

In practical LU factorization, the matrices $\mathbf{L}_k$ are never formed. The coefficients l_{jk} are computed and stored directly into $\mathbf{L}$. The complete algorithm is described in Algorithm 6.

Algorithm 6 LU factorization without pivoting

```
Set U = A and L = I
for k = 1 to n − 1 do
  for i = k + 1 to n do
    l_ik = u_ik/u_kk
    for j = k to n do
      u_i,j = u_i,j − l_ik u_k,j
    end for
  end for
end for
```

Counting the number of operations, we have

$$\sum_{k=1}^{n-1} \sum_{i=k+1}^{n} \left(1 + \sum_{j=k}^{n} 2 \right) = \sum_{k=1}^{n-1} \sum_{i=k+1}^{n} (2(n-k)+3) = \sum_{k=1}^{n-1} (2(n-k)+3)\,(n-k).$$

After changing variables, we get

$$\sum_{k'=1}^{n-1} 2k'^2 + 3k' = 2\frac{n(n-1)(2n-1)}{6} + 3\frac{n(n-1)}{2} \approx \frac{4n^3}{6} + \mathcal{O}(n^2).$$

The total number of floating point operations is $\approx \frac{2}{3}n^3$ flops.

Remark. The LU factorization can also be applied to rectangular matrices but this is rarely done in practice. So we focus on square matrices.

Remark. We described the LU factorization with an emphasis on zeroing columns. We could just as easily introduce zeros in rows. The Crout factorization takes the matrix form

$$\mathbf{A} = \tilde{\mathbf{L}}\tilde{\mathbf{U}},$$

where $\tilde{\mathbf{L}}$ is a lower triangular matrix and $\tilde{\mathbf{U}}$ is an upper triangular matrix with unit entries on the diagonal.

Remark. If the matrix $\mathbf{A}$ has an LU factorization, then we can set $\mathbf{D} = \text{diag}(\mathbf{U})$ and $\mathbf{M}^T = \mathbf{D}^{-1}\mathbf{U}$ and we write

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{M}^T,$$

where the matrix $\mathbf{D}$ is diagonal and both matrices $\mathbf{L}$ and $\mathbf{M}$ are lower triangular matrices with unit entries on the diagonal.

Remark. If the matrix $\mathbf{A}$ is symmetric, the LDM factorization becomes

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T.$$

This factorization requires $\approx \frac{n^3}{3}$ floating point operations. When all the entries on the diagonal of $\mathbf{D}$ are positive, then we can set $\mathbf{R} = \mathbf{D}^{1/2}\mathbf{L}^T$ and we obtain

$$\mathbf{A} = \mathbf{R}^T\mathbf{R},$$

which is called the Cholesky factorization. When a symmetric matrix $\mathbf{A}$ has a Cholesky factorization, it is symmetric positive definitive, *i.e.* it satisfies

$$\forall \mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\},\ \mathbf{x}^T\mathbf{A}\mathbf{x} > 0.$$

To solve a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ by LU factorization, the following steps are required:

1. Compute a LU factorization $\mathbf{A} = \mathbf{L}\mathbf{U}$ (Cost $\approx \frac{2}{3}n^3 = \mathcal{O}(n^3)$)
2. Compute $\mathbf{y} = \mathbf{L}^{-1}\mathbf{b}$ (forward substitution) (Cost $\approx n^2$)
3. Compute $\mathbf{x} = \mathbf{U}^{-1}\mathbf{y}$ (backward substitution) (Cost $\approx n^2$)

This algorithm requires less floating point operations than a QR factorization. Note that when you want to solve several linear systems with the same matrix $\mathbf{A}$ and different right hand side vectors $\mathbf{b}$, the LU factorization should be computed only once. The forward and backward substitutions require then $\mathcal{O}(n^2)$ floating point operations per right hand side.

Example 201. Determine the polynomial $p(t) = c_1 + c_2 t$ that satisfies:

$$p(1) = 2 \quad p(3) = 5.$$

The linear system associated with these equations is

$$\begin{array}{rcl} c_1 + c_2 & = & 2 \\ c_1 + 3c_2 & = & 5 \end{array}$$

Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix}.$$

We have

$$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}.$$

The LU factorization of $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}.$$

The coefficients satisfy

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

Finally, we obtain

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 3/2 \end{bmatrix}.$$

Example 202. We want to find a quadratic polynomial

$$p(t) = c_1 + c_2 t + c_3 t^2 \tag{9.7}$$

such that

$$p(2) = 0.8, \quad p(4) = 1.6, \quad p(5) = 3.2. \tag{9.8}$$

We write a linear system for the coefficients c_1, c_2, and c_3:

$$\begin{array}{rcl} c_1 + 2c_2 + 4c_3 & = & 0.8 \\ c_1 + 4c_2 + 16c_3 & = & 1.6 \\ c_1 + 5c_2 + 25c_3 & = & 3.2 \end{array}.$$

We eliminate the parameter c_1 in the last two equations to get

$$\begin{array}{rcl} c_1 + 2c_2 + 4c_3 & = & 0.8 \\ 2c_2 + 12c_3 & = & 0.8 \\ 3c_2 + 21c_3 & = & 2.4 \end{array}.$$

Now, we eliminate the coefficient c_2 in the last equation

$$\begin{aligned} c_1 + 2c_2 + 8c_3 &= 0.8 \\ 2c_2 + 12c_3 &= 0.8 \\ 3c_3 &= 1.2 \end{aligned} .$$

With backward substitution, we get

$$\begin{aligned} c_3 &= 0.4 \\ c_2 &= \tfrac{1}{2}(0.8 - 12 \times 0.4) = \tfrac{1}{2}(0.8 - 4.8) = -2 \\ c_1 &= 0.8 - 2 \times 2 + 4 \times 0.4 = 3.2 \end{aligned} .$$

So the polynomial is

$$p(t) = 3.2 - 2t + 0.4t^2.$$

Instead of using the basis $(1, t, t^2)$, we could choose the basis $(1, t-2, (t-2)(t-4))$ and write

$$p(t) = d_1 + d_2(t-2) + d_3(t-2)(t-4).$$

Using the values (9.8), the linear system for the coefficients d_1, d_2, and d_3 is

$$\begin{aligned} d_1 &= 0.8 \\ d_1 + 2d_2 &= 1.6 \\ d_1 + 3d_2 + 3d_3 &= 3.2 \end{aligned} .$$

Solving by forward substitution, we get

$$\begin{aligned} d_1 &= 0.8 \\ d_2 &= \tfrac{1}{2}(1.6 - 0.8) = 0.4 \\ d_3 &= \tfrac{1}{3}(3.2 - 3 \times 0.4 - 0.8) = 0.4 \end{aligned} .$$

So the polynomial is

$$p(t) = 0.8 + 0.4(t-2) + 0.4(t-2)(t-4).$$

9.3 LU factorization with pivoting

Unfortunately, the LU factorization as presented so far is unusable for solving general linear systems because it is not backward stable. For certain matrices, the LU factorization fails entirely because it tries to divide by zero.

Example 203. Consider the matrix

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

This matrix has full rank and is well-conditioned. Nevertheless, the LU factorization fails at the first step.

Suppose we slightly perturb the matrix

$$\mathbf{A} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}.$$

The LU factorization does not fail now. The following factors are produced

$$\mathbf{L} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}.$$

In floating point arithmetic, with 16 digits of accuracy, the numbers will be rounded to the nearest floating point number. The floating point matrices will be

$$\tilde{\mathbf{L}} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} \quad \tilde{\mathbf{U}} = \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix}.$$

These matrices are relatively close to the exact matrices. However, the product is

$$\tilde{\mathbf{L}}\tilde{\mathbf{U}} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 0 \end{bmatrix}.$$

The instability can be controlled by permuting the order of rows of the matrix being operated on. This operation is called *pivoting*.

Row permutations can also be written in terms of matrix multiplication. For example, we have

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} = \begin{bmatrix} 5 & 6 & 7 \\ 2 & 3 & 4 \\ 8 & 9 & 10 \end{bmatrix}. \tag{9.9}$$

Column permutations are also possible, for example

$$\begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 4 \\ 6 & 5 & 7 \\ 9 & 8 & 10 \end{bmatrix} \tag{9.10}$$

Note that permutation matrices are identity matrices with permuted rows. They are **orthogonal** matrices, *i.e.* they satisfy

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{T}. \tag{9.11}$$

We can also combine rows and columns permutations. For example, we have

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 5 & 6 & 7 \\ 2 & 3 & 4 \\ 8 & 9 & 10 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 6 & 5 & 7 \\ 3 & 2 & 4 \\ 9 & 8 & 10 \end{bmatrix}$$

9.3.1 Partial pivoting

At step k of LU factorization, multiples of row k are subtracted from rows $k+1, \ldots, n$ of the working matrix $\mathbf{X}$ in order to introduce zeros under the diagonal. The entry x_{kk} plays a special role and is called the pivot. Schematically, we have

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & 0 & * & * & * \\ & 0 & * & * & * \\ & 0 & * & * & * \end{bmatrix}.$$

However, there is no particular reason why the k-th row must be chosen for the elimination. For example, we could easily introduce zeros in column k by adding multiples of some row i with $k \leq i \leq n$,

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & 0 & * & * & * \\ & 0 & * & * & * \\ & x_{ik} & \times & \times & \times \\ & 0 & * & * & * \end{bmatrix}.$$

So we are free to choose as pivot any entry in the k-th column below the diagonal, as long as it is nonzero. The possibility that an entry x_{kk} might be zero implies that some flexibility in choosing the pivot may sometimes be necessary. In practice and for numerical stability, it is common to pick as pivot the largest number among the entries below the diagonal. This strategy is called *partial pivoting* (as opposed to *complete pivoting* described in Section 9.3.2).

Example 204. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}.$$

With partial pivoting, the first step is to interchange the first and third row (left-multiplication by $\mathbf{P}_1$):

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix}.$$

Then we eliminate values in the first column (left multiplication by $\mathbf{L}_1$)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{4} & 0 & 1 & 0 \\ -\frac{3}{4} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ 0 & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix}.$$

Now we permute rows 2 and 4 (left multiplication by $\mathbf{P}_2$)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ 0 & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ 0 & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}.$$

We eliminate the second column entries (left multiplication by $\mathbf{L}_2$)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{3}{7} & 1 & 0 \\ 0 & \frac{2}{7} & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ 0 & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & 0 & -\frac{2}{7} & \frac{4}{7} \\ 0 & 0 & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix}.$$

The third and fourth rows are now interchanged (left multiplication by $\mathbf{P}_3$)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & 0 & -\frac{2}{7} & \frac{4}{7} \\ 0 & 0 & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & 0 & -\frac{6}{7} & -\frac{2}{7} \\ 0 & 0 & -\frac{2}{7} & \frac{4}{7} \end{bmatrix}.$$

The final step is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & 0 & -\frac{6}{7} & -\frac{2}{7} \\ 0 & 0 & -\frac{2}{7} & \frac{4}{7} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & 0 & -\frac{6}{7} & -\frac{2}{7} \\ 0 & 0 & 0 & \frac{2}{3} \end{bmatrix}.$$

The elimination can be written in matrix form

$$\mathbf{L}_3\mathbf{P}_3\mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1\mathbf{A} = \mathbf{U}.$$

The multiplying matrix on the left side of $\mathbf{A}$ does not look like a lower triangular matrix. However, we can write

$$\mathbf{L}_3\mathbf{P}_3\mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1 = \mathbf{L}_3^{'}\mathbf{L}_2^{'}\mathbf{L}_1^{'}\mathbf{P}_3\mathbf{P}_2\mathbf{P}_1,$$

where we have

$$\mathbf{L}_3^{'} = \mathbf{L}_3, \quad \mathbf{L}_2^{'} = \mathbf{P}_3\mathbf{L}_2\mathbf{P}_3^{-1}, \quad \mathbf{L}_1^{'} = \mathbf{P}_3\mathbf{P}_2\mathbf{L}_1\mathbf{P}_2^{-1}\mathbf{P}_3^{-1}.$$

Indeed, we have

$$\mathbf{L}_3\mathbf{P}_3\mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1 = \mathbf{L}_3\mathbf{P}_3\mathbf{L}_2(\mathbf{P}_3^{-1}\mathbf{P}_3)\mathbf{P}_2\mathbf{L}_1(\mathbf{P}_2^{-1}\mathbf{P}_3^{-1}\mathbf{P}_3\mathbf{P}_2)\mathbf{P}_1.$$

The matrices $\mathbf{L}_2^{'}$ and $\mathbf{L}_1^{'}$ satisfy

$$\mathbf{L}_2^{'} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{2}{7} & 1 & 0 \\ 0 & \frac{3}{7} & 0 & 1 \end{bmatrix} \quad \mathbf{L}_1^{'} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{3}{4} & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 \\ -\frac{1}{4} & 0 & 0 & 1 \end{bmatrix}.$$

The product of the matrices $\mathbf{L}_3^{'}\mathbf{L}_2^{'}\mathbf{L}_1^{'}$ is a lower triangular matrix with unit entries on the diagonal and it is easily invertible by negating the subdiagonal entries, just as in LU factorization without pivoting. Writing $\mathbf{L} = (\mathbf{L}_3^{'}\mathbf{L}_2^{'}\mathbf{L}_1^{'})^{-1}$ and $\mathbf{P} = \mathbf{P}_3\mathbf{P}_2\mathbf{P}_1$, we have

$$\mathbf{PA} = \mathbf{LU}$$

or

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{3}{4} & 1 & 0 & 0 \\ \frac{1}{2} & -\frac{2}{7} & 1 & 0 \\ \frac{1}{4} & -\frac{3}{7} & -\frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ 0 & 0 & -\frac{6}{7} & -\frac{2}{7} \\ 0 & 0 & 0 & \frac{2}{3} \end{bmatrix}.$$

For a general $n \times n$ matrix, the factorization provided by LU factorization with partial pivoting can be written in the form

$$\mathbf{PA} = \mathbf{LU}, \tag{9.12}$$

where $\mathbf{P}$ is a permutation matrix, $\mathbf{L}$ is a lower triangular matrix with unit entries on the diagonal and lower-triangular entries smaller or equal to 1 in magnitude, and $\mathbf{U}$ is upper triangular.

The complete algorithm for LU factorization with partial pivoting is described in Algorithm 7. This algorithm requires the same number of floating point operations, $\approx \frac{2}{3}n^3$, and $\mathcal{O}(n^2)$ comparisons to find the pivots.

9.3.2 Complete pivoting

In complete pivoting, the selection of pivots is choosing the largest in the whole matrix. It requires a significant amount of time because the comparisons are now of order $\mathcal{O}(n^3)$. In practice, this is rarely done because the improvement in stability is marginal.

In matrix form, the LU factorization with complete pivoting takes the form

$$\mathbf{PAQ} = \mathbf{LU}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are permutation matrices, $\mathbf{L}$ is a lower triangular matrix with unit entries on the diagonal, and $\mathbf{U}$ is an upper triangular matrix.

9.4 Banded systems

The LU factorization requires, approximately, $\frac{2}{3}n^3$ floating point operations. Table 9.1 lists estimates of the time required to compute the LU factorization and solve a linear system on a 100 MFlops computer.

Algorithm 7 LU factorization with partial pivoting

Set $\mathbf{U} = \mathbf{A}$, $\mathbf{L} = \mathbf{I}$, and $\mathbf{P} = \mathbf{I}$
for $k = 1$ to $n - 1$ **do**
  Select $i \geq k$ to maximize $|u_{ik}|$
  for $r = k$ to n **do**
    $tmp = u_{k,r}$
    $u_{k,r} = u_{i,r}$
    $u_{i,r} = tmp$
  end for
  for $r = 1$ to $k - 1$ **do**
    $tmp = l_{k,r}$
    $l_{k,r} = l_{i,r}$
    $l_{i,r} = tmp$
  end for
  for $r = 1$ to n **do**
    $tmp = p_{k,r}$
    $p_{k,r} = p_{i,r}$
    $p_{i,r} = tmp$
  end for
  for $j = k + 1$ to n **do**
    $l_{jk} = u_{jk}/u_{kk}$
    for $r = k$ to n **do**
      $u_{j,r} = u_{j,r} - l_{jk}u_{k,r}$
    end for
  end for
end for

n	Flops	Time
10	6.6×10^2	6.6×10^{-6} s
100	6.6×10^5	6.6×10^{-3} s
1000	6.6×10^8	6.6×10^0 s
10000	6.6×10^{11}	6.6×10^3 s $\approx$ 111.11 min
100000	6.6×10^{14}	6.6×10^6 s $\approx$ 1850 h $\approx$ 77 days
1000000	6.6×10^{17}	6.6×10^9 s $\approx$ 77000 days $\approx$ 210 years

Table 9.1: Time estimates to solve $\mathbf{Ax} = \mathbf{b}$ by LU factorization on a 100 MFlops computer

We assume that all the problems fit in the memory and that the initial matrix is dense (all the entries could be nonzero). Luckily, many practical systems have some structure or involve *sparse* matrices (where most elements are zero). An example of a sparse matrix is a diagonal matrix. Solving $\mathbf{Ax} = \mathbf{b}$, when $\mathbf{A}$ is a diagonal matrix, requires only $\mathcal{O}(n)$ flops and this takes only 0.005 seconds rather than 210 years!

Banded systems are another examples of matrices for which the LU factorization requires only $\mathcal{O}(n)$ flops. A matrix is banded whenever the equations can be ordered so that each unknown x_i appears in only a few equations in a "neighborhood" of the i-th equation. Formally, we say that $\mathbf{A}$ has upper bandwidth q if $a_{ij} = 0$ whenever $j > i+q$ and lower bandwidth p if $a_{ij} = 0$ whenever $i > j + p$.

Theorem 205. *Suppose $\mathbf{A} \in \mathbb{R}^{n\times n}$ has an LU factorization $\mathbf{A} = \mathbf{LU}$. If $\mathbf{A}$ has upper bandwidth q and lower bandwidth p, then $\mathbf{U}$ has upper bandwidth q and $\mathbf{L}$ has lower bandwidth p.*

The proof is given in Golub and Van Loan, *"Matrix Computations"*, The John Hopkins University Press, 1996. Then the LU factorization can be specialized and will involve only $2npq$ flops. The forward substitution requires about $2np$ flops and the backward substitution about $2nq$ flops. Note that, however, pivoting will destroy the band structure.

Example 206. Consider a $n\times n$ symmetric, tridiagonal, positive definite matrix $\mathbf{A}$. Setting

$$\mathbf{L} = \begin{bmatrix} 1 & & & \dots & 0 \\ l_1 & 1 & & & \vdots \\ & \ddots & \ddots & & \\ \vdots & & & 1 & \\ 0 & \dots & & l_{n-1} & 1 \end{bmatrix}$$

and $\mathbf{D} = \text{diag}(d_1, \ldots, d_n)$, the equation $\mathbf{A} = \mathbf{LDL}^T$ gives

$$\begin{aligned} a_{11} &= d_1 \\ a_{k,k-1} &= l_{k-1}d_{k-1} \\ a_{k,k} &= d_k + l_{k-1}^2 d_{k-1} = d_k + l_{k-1}a_{k,k-1} \end{aligned}$$

The complete algorithm to compute the factorization and to solve a linear system $8n$ flops.

Example 207. Consider the mass spring system in Figure 9.1.

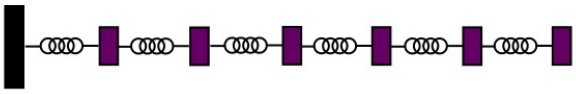


Figure 9.1: Mass-spring system

Newton's law $\mathbf{f} = m\mathbf{a}$ applied to this system yields

$$\begin{aligned} m_i\ddot{x}_i &= k_i(x_{i-1} - x_i) \quad \text{(force on mass } i \text{ from spring } i\text{)} \\ &\quad +k_{i+1}(x_{i+1} - x_i) \quad \text{(force on mass } i \text{ from spring } i+1\text{)} \\ &\quad +f_i \quad \text{(external force on mass } i\text{)} \end{aligned}$$

or

$$\mathbf{M}\ddot{\mathbf{x}} = -\mathbf{K}\mathbf{x} + \mathbf{f}$$

where the matrix $\mathbf{M}$ is diagonal and the matrix $\mathbf{K}$ is tridiagonal. $\mathbf{M}$ is called the mass matrix and $\mathbf{K}$ is called the stiffness matrix. Electrical engineers analyzing circuits arrive at an analogous equation by applying Kirchoff's and related laws instead of Newton's law. In this case, $\mathbf{x}$ represents branch currents, $\mathbf{M}$ represents inductances, and $\mathbf{K}$ represents admittances.

9.5 Useful commands in MATLAB

Here are a few commands in MATLAB useful for this section.

- `inv(A)` computes the inverse of matrix $\mathbf{A}$ when it exists.
- `[L,U,P]=lu(A)` returns an upper triangular matrix in $\mathbf{U}$, a lower triangular matrix $\mathbf{L}$ with a unit diagonal, and a permutation matrix $\mathbf{P}$, such that $\mathbf{LU} = \mathbf{PA}$.
- If the matrix $\mathbf{A}$ is square, `A\b` is roughly the same as `inv(A)*b`, except it is more efficiently.
  - It checks whether $\mathbf{A}$ is diagonal, banded, triangular, a permutation of a triangular.

– If **A** is symmetric, use the Cholesky or the $\mathbf{LDL}^T$ factorization
– Use the LU factorization.
– If **A** is not square, use the QR factorization
– For further information, look at "`help mldivide`".

10 Least squares problem

The term *least squares* describes a frequently used approach to solving overdetermined or inexactly specified systems of equations in an approximate sense. Instead of solving the equations exactly, we seek only to minimize the sum of the squares of the residuals.

10.1 Residuals and norms

Suppose the linear system $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A} \in \mathbb{R}^{m\times n}$ has no solution because $\mathbf{b}$ is not in the range of $\mathbf{A}$. There are many situations where this arises but we still want to find a vector $\mathbf{x} \in \mathbb{R}^n$ that approximately solves the system. The *residual vector* $\mathbf{r} \in \mathbb{R}^m$ is defined by

$$\mathbf{r} = \mathbf{b} - \mathbf{Ax}.$$

If $\mathbf{x}$ solves the system exactly then $\mathbf{r} = \mathbf{0}$. Otherwise we can use some vector norm to measure how small the residual is, $\|\mathbf{r}\| = \|\mathbf{b} - \mathbf{Ax}\|$. For any particular $\mathbf{x} \in \mathbb{R}^n$, this will be a nonnegative real number.

We say that a particular vector $\mathbf{x}^* \in \mathbb{R}^n$ is the "best approximation" in some norm if

$$\|\mathbf{b} - \mathbf{Ax}^*\| \le \|\mathbf{b} - \mathbf{Ax}\|, \quad \forall \mathbf{x} \in \mathbb{R}^n.$$

The "best" solution will depend on what norm we use.

If we use the 2-norm to measure the residual then

$$\|\mathbf{b} - \mathbf{Ax}\|_2 = \sqrt{\sum_{i=1}^{m} (b_i - (\mathbf{Ax})_i)^2} \tag{10.1}$$

where $(\mathbf{Ax})_i$ means the i-th component of the vector $\mathbf{Ax} \in \mathbb{R}^m$. In this case, we are trying to find the vector $\mathbf{x}$ that minimizes the sum of the squares of the residuals, and the vector $\mathbf{x}^*$ that minimizes this is called the least squares solution to the linear system.

Remark. Minimizing the sum of squares or the square root of the sum of squares gives the same minimizer $\mathbf{x}^*$ since the square root is an increasing function. So we can equally well minimize

$$\|\mathbf{b} - \mathbf{Ax}\|_2^2 = \sum_{i=1}^{m} (b_i - (\mathbf{Ax})_i)^2 \tag{10.2}$$

The vector $\mathbf{x}^*$ that minimizes (10.2) will be the same as the vector that minimizes (10.1).

If some observations are more important or more accurate than others, then we might associate different weights, ω_j, with different observations and minimize

$$\|\mathbf{r}\|_\omega^2 = \sum_{i=1}^{m} \omega_i r_i^2.$$

With the 1-norm, we minimize the sum of the absolute values of the residuals:

$$\|\mathbf{r}\|_1 = \sum_{i=1}^{m} |r_i|.$$

This problem can be reformulated as a linear programming problem, but it is computationally more difficult than the least squares problem. The resulting solutions are less sensitive the presence of spurious data points or *outliers*.

With the ∞-norm, we minimize the largest residual:

$$\|\mathbf{r}\|_\infty = \max_i |r_i|.$$

This is also known as a Chebyshev fit and can be reformulated as a linear programming problem. Chebyshev fits are frequently used in the design of digital filtes and in the development of approximations for use in mathematical function libraries.

We will limit ourselves to least squares problem in this class.

Remark. We will generally assume $m > n$ and that $\text{rank}(\mathbf{A}) = n$ so that the columns of $\mathbf{A}$ are linearly independent. In this case it can be shown that the least squares solution is unique. (If $\text{rank}(\mathbf{A}) < n$, we could add a nontrivial element of the null space of $\mathbf{A}$ to $\mathbf{x}^*$ and get another solution.)

10.2 Examples of least squares problem

First we present some examples of least squares problems. In MATLAB, it is easy to solve a linear least squares problem using the backslash operator. In this lecture, we will see what's going on under the hood when you do this.

Example 208. Consider the matrix $\mathbf{A} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$ and the vector $\mathbf{b} = \begin{bmatrix} 6 \\ 7 \\ 8 \end{bmatrix}$. We want to find the value of x that minimizes the length of the vector $\mathbf{A}[x] - \mathbf{b}$, measured in the 2-norm. Using the backslash operator from MATLAB, we get $x = 2$. Note that, geometrically, the value of x is the one that makes the vector $\mathbf{A}[x] - \mathbf{b}$ perpendicular to the column vector $\mathbf{A}$

$$\begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}^T \left(\begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} \times 2 - \begin{bmatrix} 6 \\ 7 \\ 8 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}^T \begin{bmatrix} -4 \\ -3 \\ 2 \end{bmatrix} = -4 - 6 + 10 = 0.$$

Example 209. Consider the matrix $\mathbf{A}$ and the vector $\mathbf{b}$

$$\mathbf{A} = \begin{bmatrix} -1 & 1 \\ 0 & 2 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} -2 \\ 6 \\ 8 \end{bmatrix}.$$

Using the backslash operator from MATLAB, we get $\mathbf{x} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$. The residual vector is

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x} = \begin{bmatrix} -2 \\ 6 \\ 8 \end{bmatrix} - \begin{bmatrix} -1 & 1 \\ 0 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ 6 \\ 8 \end{bmatrix} - \begin{bmatrix} -1 \\ 4 \\ 9 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}.$$

Note that, now, the residual vector is orthogonal to the column vectors of $\mathbf{A}$

$$\begin{bmatrix} -1 & 1 \\ 0 & 2 \\ 1 & 3 \end{bmatrix}^T \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 1-1 \\ -1+4-3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Example 210. Find the quadratic function $p(t) = c_1 + c_2 t + c_3 t^2$ that gives a least squares fit to the data $(t_{i,}, y_i)$ from the table below

i	t_i	y_i
1	0	5
2	1	4
3	2	6
4	3	9

If we try to require $p(t_i) = y_i$, we get 4 equations in the 3 unknowns c_1, c_2, and c_3, which has the form

$$\begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \\ 1 & t_4 & t_4^2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}.$$

From the data above, the matrix $\mathbf{A}$ and the input vector $\mathbf{y}$ are

$$\mathbf{y} = \begin{bmatrix} 5 \\ 4 \\ 6 \\ 9 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix}$$

Note that if we only required 3 of these conditions we would get a 3×3 square system with a unique solution, corresponding to a parabola going through those three points. But there is no single quadratic function that goes through all four points given in the table above, and the vector $\mathbf{y}$ does not lie in the column space of the matrix $\mathbf{A}$. Using the backslash operator in MATLAB, we get

$$\mathbf{c} = \begin{bmatrix} 4.9 \\ -1.6 \\ 1 \end{bmatrix}.$$

Figure 10.1 plots the data and the least-squares fit

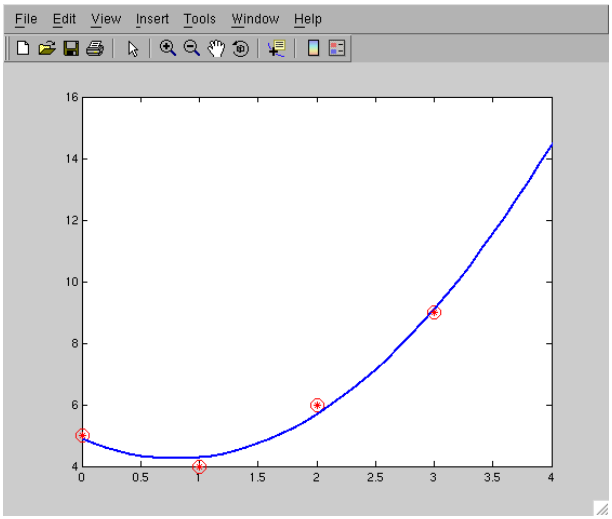


Figure 10.1: Least-squares fit with a quadratic function

10.3 Normal equations

If we want a least squares solution to a linear system $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$ and $\text{rank}(\mathbf{A}) = n$ (the columns of $\mathbf{A}$ are linearly independent), then the unique least squares solution can be found by solving the square $n \times n$ system

$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}. \tag{10.3}$$

These equations are called the normal equations. It can be shown that if $\text{rank}(\mathbf{A}) = n$ then $\text{rank}(\mathbf{A}^T\mathbf{A}) = n$ as well, so this matrix is nonsingular and the system has a unique solution.

There are several ways to derive the normal equations. The best way to remember where these equations come from is to rewrite (10.4) as

$$\mathbf{A}^T(\mathbf{Ax} - \mathbf{b}) = \mathbf{0}. \tag{10.4}$$

Recall that the vector $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$ is the residual, and (10.4) states that the residual vector is orthogonal to all columns of $\mathbf{A}$, and hence the residual is orthogonal to $\mathcal{R}(\mathbf{A})$, the space spanned by the columns of $\mathbf{A}$.

Example 211. Consider the 2×1 least squares problem

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} x = \begin{bmatrix} 3 \\ 1 \end{bmatrix}. \tag{10.5}$$

The right hand side $\mathbf{b}$ does not lie in the column space of $\mathbf{A}$, which is the line in 2-space through the origin and the point $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$. The solution to the least

squares problem is the scalar x for which $\begin{bmatrix} 2x \\ x \end{bmatrix}$ (a point on the line) is closest (in Euclidean distance) to the point $\begin{bmatrix} 3 \\ 1 \end{bmatrix}$. It is obvious from Figure 10.2 that this is the point for which the residual is orthogonal to the line.

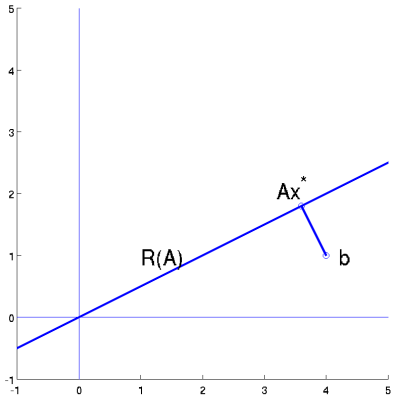


Figure 10.2: Solution to the least squares problem (10.5)

Another way to derive the normal equations is to minimize the function

$$r(\mathbf{x}) = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = (\mathbf{b} - \mathbf{A}\mathbf{x})^T (\mathbf{b} - \mathbf{A}\mathbf{x}) .$$

This function is a smooth function of $\mathbf{x}$ and is quadratic in the components of $\mathbf{x}$. To find the minimum for r, we can set $r'(\mathbf{x}) = \mathbf{0}$. To find r', we recall that the derivative is the first linear term in the Taylor expansion. So we compute

$$\begin{aligned}
r(\mathbf{x}+\mathbf{y}) &= (\mathbf{b} - \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y})^T(\mathbf{b} - \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y}) \\
r(\mathbf{x}+\mathbf{y}) &= (\mathbf{b} - \mathbf{A}\mathbf{x})^T(\mathbf{b} - \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y}) - \mathbf{y}^T\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y}) \\
r(\mathbf{x}+\mathbf{y}) &= (\mathbf{b} - \mathbf{A}\mathbf{x})^T(\mathbf{b} - \mathbf{A}\mathbf{x}) - (\mathbf{b} - \mathbf{A}\mathbf{x})^T\mathbf{A}\mathbf{y} - (\mathbf{A}\mathbf{y})^T(\mathbf{b} - \mathbf{A}\mathbf{x}) + (\mathbf{A}\mathbf{y})^T\mathbf{A}\mathbf{y} \\
r(\mathbf{x}+\mathbf{y}) &= r(\mathbf{x}) - 2\mathbf{y}^T\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}) + (\mathbf{A}\mathbf{y})^T\mathbf{A}\mathbf{y}
\end{aligned}$$

So we obtain

$$r'(\mathbf{x}) = 2\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) = 2\left(\mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}^T\mathbf{b}\right)$$

which gives the normal equation when setting it to $\mathbf{0}$.

Note that because $r(\mathbf{x})$ is quadratic in the components of $\mathbf{x}$, setting the derivatives to 0 gives a linear systems of equations. This is why minimizing the 2-norm of residuals is easy to solve using linear algebra. In other norms, it is not so easy. For example, with the 1-norm, the function r is not smooth and we can not find the minimum by setting the derivative equal to 0.

10.3.1 Classical solution

The classical way to solve least-squares problems is to solve the normal equations. We assume that the matrix $\mathbf{A}$ is full rank. The matrix $\mathbf{A}^T\mathbf{A}$ is square, symmetric positive definite. The standard method of solving is by a Cholesky factorization. It constructs a factorization $\mathbf{A}^T\mathbf{A} = \mathbf{R}^T\mathbf{R}$, where $\mathbf{R}$ is upper triangular. Algorithm 8 solves the least squares problems.The number of float-

Algorithm 8 Least squares via normal equations

Form the matrix $\mathbf{A}^T\mathbf{A}$ and the vector $\mathbf{A}^T\mathbf{b}$.
Compute the Cholesky factorization $\mathbf{A}^T\mathbf{A} = \mathbf{R}^T\mathbf{R}$.
Solve the lower triangular system $\mathbf{R}^T\mathbf{w} = \mathbf{A}^T\mathbf{b}$.
Solve the upper triangular system $\mathbf{R}\mathbf{x} = \mathbf{w}$.

ing point operations is, approximately, $mn^2 + \frac{n^3}{3}$. However, if κ denotes the condition number of matrix $\mathbf{A}$, then the matrix $\mathbf{A}^T\mathbf{A}$ has condition number κ^2. Thus the best we can expect from the normal equations is

$$\frac{\|\mathbf{x} - \mathbf{x}_{computed}\|}{\|\mathbf{x}\|} = \mathcal{O}(\kappa^2 \epsilon_{\text{machine}}).$$

10.3.2 QR factorization

The modern method for solving least squares problems is based upon reduced QR factorization. One constructs a factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$. So we write the normal equations $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$ and we have

$$\hat{\mathbf{R}}^T\hat{\mathbf{Q}}^T\hat{\mathbf{Q}}\hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{R}}^T\hat{\mathbf{Q}}^T\mathbf{b} \;\Rightarrow\; \hat{\mathbf{R}}^T\hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{R}}^T\hat{\mathbf{Q}}^T\mathbf{b} \;\Rightarrow\; \hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{Q}}^T\mathbf{b}.$$

This last equation is upper triangular and $\hat{\mathbf{R}}$ is nonsingular because matrix $\mathbf{A}$ is full rank. The steps for computing the solution are described in Algorithm 9.The cost is dominated by the QR factorization. Recall that the MATLAB

Algorithm 9 Least squares via reduced QR factorization

Compute the reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$.
Compute the vector $\hat{\mathbf{Q}}^T\mathbf{b}$.
Solve the upper triangular system $\hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{Q}}^T\mathbf{b}$.

function `qr` requires asymptotically $2mn^2 - \frac{2}{3}n^3$ flops.

10.3.3 SVD factorization

Another method for solving least squares problems uses the SVD decomposition. This decomposition is a very useful result.

Theorem 212. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \geq n$. Then we can write*

$$\mathbf{A} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^T,$$

where $\hat{\mathbf{U}}$ *is of dimension* $m \times n$ *and satisfies* $\hat{\mathbf{U}}^T\hat{\mathbf{U}} = \mathbf{I}$, $\mathbf{V}$ *is of dimension* $n \times n$ *and satisfies* $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, *and* $\hat{\mathbf{\Sigma}}$ *is a* $n \times n$ *diagonal matrix* $\text{diag}(\sigma_1, \ldots, \sigma_n)$, *where* $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$. *The column vectors of* $\hat{\mathbf{U}}$ *are called left singular vectors. The column vectors of* $\mathbf{V}$ *are called right singular vectors. The scalar* σ_i *are called singular values.*

For the least squares problem, we write the normal equations $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$ and we obtain

$$\mathbf{V}\hat{\mathbf{\Sigma}}\hat{\mathbf{U}}^T\hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^T\mathbf{x} = \mathbf{V}\hat{\mathbf{\Sigma}}\hat{\mathbf{U}}^T\mathbf{b} \;\Rightarrow\; \hat{\mathbf{\Sigma}}\mathbf{V}^T\mathbf{x} = \hat{\mathbf{U}}^T\mathbf{b}.$$

The matrix $\hat{\mathbf{\Sigma}}$ is nonsingular when the matrix $\mathbf{A}$ is full rank. The steps for computing the solution are described in Algorithm 10.The cost is dominated

Algorithm 10 Least squares via SVD decomposition

Compute the reduced SVD decomposition $\mathbf{A} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^T$.
Compute the vector $\hat{\mathbf{U}}^T\mathbf{b}$.
Solve the diagonal system $\hat{\mathbf{\Sigma}}\mathbf{w} = \hat{\mathbf{U}}^T\mathbf{b}$.
Set $\mathbf{x} = \mathbf{V}\mathbf{w}$.

by the SVD decomposition. The MATLAB function `svd` requires asymptotically $2mn^2 + 11n^3$ flops. This method is useful when the matrix $\mathbf{A}$ is close to rank-deficient.

10.4 Approximation of functions

Given some function $f(t)$, we want to find a function $g(t)$ from some finite dimensional function space that approximates $f(t)$ over some interval $[a, b]$. For example, we can choose $g \in \mathcal{P}_k$, a polynomial of degree k for some k.

Recall that a function space of dimension n can be described by n basis functions $\phi_1(t), \ldots, \phi_n(t)$. The problem is to find the coefficients $c_1, \ldots, c_n$ in

$$g(t) = c_1\phi_1(t) + \ldots + c_n\phi_n(t).$$

The best choice of "basis functions" $\phi_j(t)$ depends on where the function f or the data comes from. In some cases, we may have data that we think comes from a linear function,

$$g(t) = c_1 + c_2 t$$

in which case we would naturally choose $n = 2$, $\phi_1(t) = 1$ and $\phi_2(t) = t$. Or if we want to fit by a quadratic we would use these functions along with $\phi_3(t) = t^2$.

For some kinds of approximation, this problem can reduce to a linear algebra problem for vector

$$\mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}.$$

In the following, we will discuss some approaches.

10.4.1 Interpolation

Given a set of n data points (t_i, y_i), we assume that the data should approximately satisfy $y = g(t)$ where $g(t)$ is a function that is a linear combination of n given functions, $\phi_1(t),\ \ldots,\ \phi_n(t)$:

$$g(t) = c_1\phi_1(t) + c_2\phi_2(t) + \cdots + c_n\phi_n(t).$$

Given the "basis functions" $\phi_j(t)$ and the data, we can solve the problem of finding the coefficients c_j to minimize

$$\sum_{i=1}^{n}(y_i - g(t_i))^2$$

by solving the linear least squares problem $\mathbf{Ac} = \mathbf{y}$, where the matrix $\mathbf{A}$ and the data vector $\mathbf{y}$ are

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} \phi_1(t_1) & \ldots & \phi_n(t_1) \\ \phi_1(t_2) & & \phi_n(t_2) \\ \vdots & & \vdots \\ \phi_1(t_n) & \ldots & \phi_n(t_n) \end{bmatrix}.$$

With reasonable choice of basis functions, this system will usually be nonsingular.

Example 213. Consider n distinct points $t_1, \ldots, t_n$ and data $y_1, \ldots, y_n$. There exists a unique polynomial interpolant to these data in these points, that is, a polynomial of degree at most $n-1$,

$$p(x) = c_0 + c_1x + \ldots + c_{n-1}x^{n-1},$$

with the property $p(t_i) = y_i$ for each index i. The relationship among the points, the data, and the coefficients can be expressed by a linear system

$$\begin{bmatrix} 1 & t_1 & \ldots & t_1^{n-1} \\ 1 & t_2 & & t_2^{n-1} \\ \vdots & & & \vdots \\ 1 & t_n & \ldots & t_n^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

We can solve this system of equations, which is guaranteed to be nonsingular as long as the points $\{t_i\}$ are distinct. Figure 10.3 presents an example of polynomial interpolation with a polynomial of degree 10 and 11 data points.

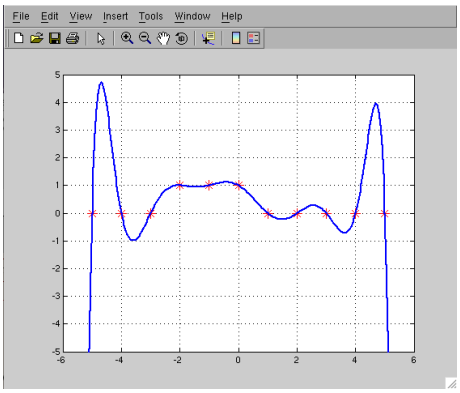


Figure 10.3: Degree 10 polynomial interpolant to 11 data points.

The interpolant passes through all the data points. However, the fit is not pleasing. Near the ends of the interval, p exhibits large oscillations that are an artifact of the interpolation, not a reflection of the data. This unsatisfactory behavior is typical of polynomial interpolation. The produced fits are often bad and they tend to get worse if more data are utilized. Even if the fit is good, the interpolation process may be ill-conditioned, *i.e.* sensitive to perturbations of the data.

Note that the structure of the system and ease of solution depends on the choice of basis functions.

Example 214. Find a polynomial $u \in \mathcal{P}_2$ interpolating $f(t) = \sqrt{t}$ at $t_i = 1, 4, 9$. Instead of using the canonical basis $(1, t, t^2)$, we select the basis

$$((t-4)(t-9), (t-1)(t-9), (t-1)(t-4)).$$

Then the system is

$$\begin{bmatrix} 24 & 0 & 0 \\ 0 & -15 & 0 \\ 0 & 0 & 40 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

The solution is

$$u(t) = \frac{1}{24}(t-4)(t-9) - \frac{2}{15}(t-1)(t-9) + \frac{3}{40}(t-1)(t-4).$$

10.4.2 Data fitting

Given a set of m data points (t_i, y_i), we assume that the data should approximately satisfy $y = g(t)$ where $g(t)$ is a function that is a linear combination of n given functions, $\phi_1(t), \ \ldots, \ \phi_n(t)$:

$$g(t) = c_1\phi_1(t) + c_2\phi_2(t) + \cdots + c_n\phi_n(t).$$

Typically, we have $m > n$.

Given the "basis functions" $\phi_j(t)$ and the data, we can solve the problem of finding the coefficients c_j to minimize

$$\sum_{i=1}^{m} (y_i - g(t_i))^2$$

by solving the linear least squares problem $\mathbf{Ac} = \mathbf{y}$, where the matrix $\mathbf{A}$ and the data vector $\mathbf{y}$ are

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} \phi_1(t_1) & \dots & \phi_n(t_1) \\ \phi_1(t_2) & & \phi_n(t_2) \\ \vdots & & \vdots \\ \phi_1(t_m) & \dots & \phi_n(t_m) \end{bmatrix}.$$

Here, again, the choice of basis functions governs the ease of solving the problem. For example, if columns of $\mathbf{A}$ are orthogonal to one another, then $\mathbf{A}^T\mathbf{A}$ is diagonal.

Example 215. Consider again the 11 data points used previously for the polynomial interpolation. The fit can be improved by reducing the degree of the polynomial and solve a least-squares problem. Figure 10.4 plots the data and the least-squares fit

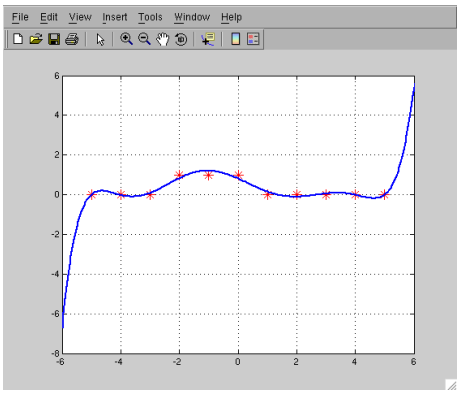


Figure 10.4: Degree 7 polynomial least squares fit to 11 data point

The new polynomial does not interpolate the data but it captures their overall behavior much better than the polynomial of degree 10. The computation is also less sensitive to perturbations.

Example 216. Suppose we want to fit the data

i	t_i	y_i
1	0	2
2	1	4
3	4	2
4	9	-4

by a function of the form $g(t) = c_1\sqrt{t} + c_2(t-1)$. The linear system would be

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 2 & 3 \\ 3 & 8 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 2 \\ -4 \end{bmatrix}.$$

The first column is the basis function $\phi_1(t) = \sqrt{t}$ evaluated at the data points and the second column is the basis function $\phi_2(t) = (t-1)$ evaluated at the data points. In this case the right hand side happens to lie in the column space of the matrix, and there is a solution $c_1 = 4$, $c_2 = -2$ that has zero residual. In other words the 4 data points all lie on the curve $g(t) = 4\sqrt{t} - 2(t-1)$. If the data $\mathbf{y}$ is changed, however, this would probably not be true and then the least squares solution would give the fit minimizing the residual. Figure 10.5 plots the data and the least-squares fit

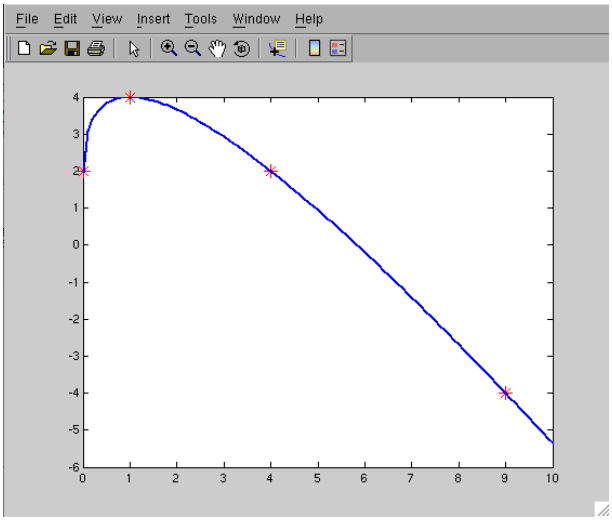


Figure 10.5: Least-squares fit with a polynomial and the square root function

10.4.3 Global approximation

Instead of fitting m data points, we can try to find a function $g(t)$ that is as close as possible to $f(t)$ at all points in $[a, b]$. Now we must measure the residual

(difference between $g(t)$ and $f(t)$) in some function norm $\|f-g\|$. For example, the norm

$$\|f\|_\infty = \max_{a\le t\le b} |f(t)|$$

is a natural generalization of the ∞-norm in $\mathbb{R}^n$. The inner product

$$< u, v >= \int_a^b u(t)v(t)dt \tag{10.6}$$

is a natural generalization of the Euclidian inner product in $\mathbb{R}^n$. The associated norm

$$\|u\|_2 = \sqrt{< u, u >} = \sqrt{\int_a^b |u(t)|^2 dt}$$

is a generalization to functions of the Euclidian norm in $\mathbb{R}^n$.

For example, assume that

$$g(t) = c_1\phi_1(t) + c_2\phi_2(t)$$

and that we are trying to minimize $\|f-g\|_2$, *i.e.*

$$\int_a^b (f(t) - c_1\phi_1(t) - c_2\phi_2(t))^2 dt.$$

We expand the equation and gather the coefficients c_1 and c_2

$$c_1^2 \int_a^b \phi_1(t)^2 dt + 2c_1c_2 \int_a^b \phi_1(t)\phi_2(t)dt + c_2^2 \int_a^b \phi_2(t)^2 dt \\ - 2c_1 \int_a^b f(t)\phi_1(t)dt - 2c_2 \int_a^b f(t)\phi_2(t)dt + \int_a^b f(t)^2 dt.$$

To compute the minimum for the norm $\|f-g\|_2$, we look for a stationary point, *i.e.* we write that the derivatives with respect to c_1 and c_2 are equal to zero

$$\begin{array}{rcl} \int_a^b f(t)\phi_1(t)dt - c_1 \int_a^b \phi_1(t)^2 dt - c_2 \int_a^b \phi_1(t)\phi_2(t)dt &=& 0 \\ \int_a^b f(t)\phi_2(t)dt - c_1 \int_a^b \phi_1(t)\phi_2(t)dt - c_2 \int_a^b \phi_2(t)^2 dt &=& 0 \end{array}.$$

The system looks nicer if we use the inner product notation

$$\begin{array}{rcl} < \phi_1, \phi_1 > c_1 + < \phi_1, \phi_2 > c_2 &=& < f, \phi_1 > \\ < \phi_2, \phi_1 > c_1 + < \phi_2, \phi_2 > c_2 &=& < f, \phi_2 > \end{array}.$$

The system has the form $\mathbf{Bc} = \mathbf{y}$ where

$$\mathbf{B} = \begin{bmatrix} < \phi_1, \phi_1 > & < \phi_1, \phi_2 > \\ < \phi_2, \phi_1 > & < \phi_2, \phi_2 > \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} < f, \phi_1 > \\ < f, \phi_2 > \end{bmatrix}. \tag{10.7}$$

These are the normal equations. When using n basis functions, the least squares problem becomes $\mathbf{Bc} = \mathbf{y}$, where the matrix $\mathbf{B}$ and the data vector $\mathbf{y}$ are

$$\mathbf{y} = \begin{bmatrix} < f, \phi_1 > \\ \vdots \\ < f, \phi_n > \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} < \phi_1, \phi_1 > & \dots & < \phi_1, \phi_n > \\ \vdots & & \vdots \\ < \phi_n, \phi_1 > & \dots & < \phi_n, \phi_n > \end{bmatrix}.$$

Note that the system (10.7) becomes diagonal if $< \phi_1, \phi_2 >= 0$ or if we choose an orthogonal basis (orthogonal for the inner product (10.6)). For example, the Legendre polynomials are a family of orthogonal polynomials on the interval $[-1, -1]$ for the inner product (10.6). The first four Legendre polynomials are

$$\begin{aligned} P_0(t) &= 1 \\ P_1(t) &= t \\ P_2(t) &= \frac{3}{2}t^2 - \frac{1}{2} \\ P_3(t) &= \frac{5}{2}t^3 - \frac{3}{2}t \end{aligned}$$

Example 217. Consider the function $f(t) = \cos(t)$ on $[-1, 1]$. We look for the best fitting in $\mathcal{P}_1$. The basis $\phi_1(t) = 1$ and $\phi_2(t) = t$ is orthogonal. We have

$$\begin{array}{lll} < \phi_1, \phi_1 >= 2 & < f, \phi_1 >= 2\sin(1) & \Longrightarrow \quad c_1 = \sin(1) \approx 0.8415 \\ < \phi_2, \phi_2 >= \frac{2}{3} & < f, \phi_2 >= 0 & \Longrightarrow \quad c_2 = 0 \end{array}$$

Figure 10.6 plots the function f and the best approximation in $\mathcal{P}_1$.

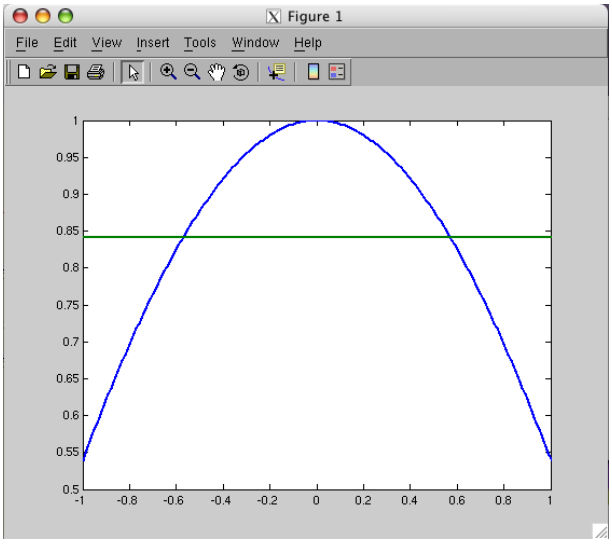


Figure 10.6: Approximation of cosine function on $[-1, 1]$ with a constant function.

Next we look for the best fitting in $\mathcal{P}_2$. The basis $\phi_1(t) = 1$, $\phi_2(t) = t$, and $\phi_3(t) = \frac{3}{2}(t^2 - \frac{1}{3})$ are orthogonal. We have

$$\begin{array}{lll} < \phi_1, \phi_1 >= 2 & < f, \phi_1 >= 2\sin(1) & \Longrightarrow \quad c_1 = \sin(1) \approx 0.8415 \\ < \phi_2, \phi_2 >= \frac{2}{3} & < f, \phi_2 >= 0 & \Longrightarrow \quad c_2 = 0 \\ < \phi_3, \phi_3 >= \frac{3}{5} & < f, \phi_3 > \approx -0.124 & \Longrightarrow \quad c_3 \approx -0.31 \end{array}$$

Figure 10.7 plots the function f and the best approximation in $\mathcal{P}_2$.

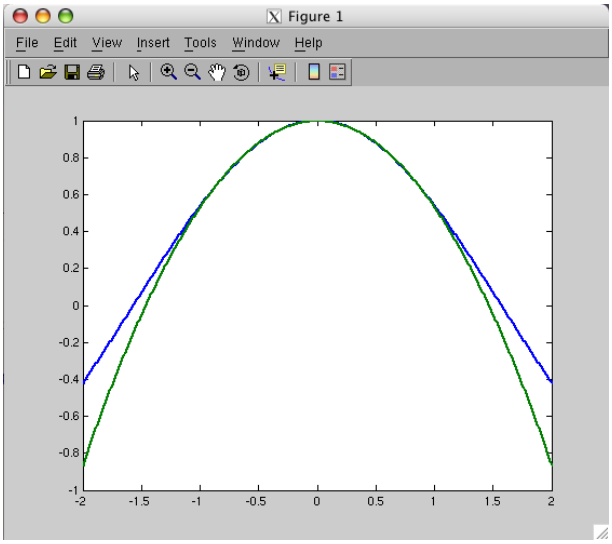


Figure 10.7: Approximation of cosine function on $[-1, 1]$ with a quadratic function.

10.5 Useful commands in MATLAB

In MATLAB, the normal equations could be formed and solved by the single command

```
>> x = (A'*A) \ A'*b;
```

Computationally, it is better to use the reduced QR factorization

```
>> [Q, R] = qr(A, 0);
>> x = R \ (Q' * b);
```

It is also possible to set

```
>> x = A \ b;
```

Other useful commands in MATLAB for this section include:

- `[U,S,V] = svd(A,0)` produces the reduced SVD decomposition. If `A` is $m \times n$ with $m \geq n$, then `svd` computes only the first n columns of $\hat{\mathbf{U}}$ and $\mathbf{\Sigma}$ is $n \times n$.
- `p = polyfit(x,y,n)` finds the coefficients of a polynomial $p(x)$ of degree n that fits the data, $p(x_i)$ to y_i, in a least squares sense.

10.6 Exercises

11 Eigenvalues and eigenvectors

In talking about eigenvalues we are only interested in square matrices, $\mathbf{A} \in \mathbb{R}^{n\times n}$. Then $\mathbf{f}(\mathbf{u}) = \mathbf{A}\mathbf{u}$ defines a mapping from $\mathbb{R}^n$ to itself. If we multiply an arbitrary vector $\mathbf{u}$ by $\mathbf{A}$ it will get mapped to a new vector $\mathbf{A}\mathbf{u}$ that is typically linearly independent from $\mathbf{u}$.

Example 218. If

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} \tag{11.1}$$

then

$$\mathbf{A}\mathbf{u} = \begin{bmatrix} 12 \\ -2 \end{bmatrix}$$

and the vector $\mathbf{A}\mathbf{u}$ is not a scalar multiple of $\mathbf{u}$, so the two are linearly independent.

However, there may be particular vectors $\mathbf{v}$ for which $\mathbf{A}\mathbf{v}$ is just a scalar multiple of $\mathbf{v}$. For the above matrix (11.1), we have

$$\mathbf{v} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \Longrightarrow \quad \mathbf{A}\mathbf{v} = \begin{bmatrix} 3 \\ 0 \end{bmatrix} = 3 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{11.2}$$

Such a vector is called an *eigenvector* of the matrix $\mathbf{A}$. Of course $\mathbf{x} = \mathbf{0}$ always has this property since $\mathbf{A}\mathbf{0} = \mathbf{0}$, but we are interested in nonzero eigenvectors of the matrix.

Definition 219. The vector $\mathbf{v} \neq \mathbf{0}$ is an *eigenvector* of $\mathbf{A}$ if $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ for some scalar value λ. The scalar λ is then called an *eigenvalue*.

If $\mathbf{v}$ is an eigenvector than so is $\alpha\mathbf{v}$ for any scalar α, since

$$\mathbf{A}(\alpha\mathbf{v}) = \alpha\mathbf{A}\mathbf{v} = \alpha(\lambda\mathbf{v}) = \lambda(\alpha\mathbf{v}).$$

More generally, if $\mathbf{u}$ and $\mathbf{v}$ are two eigenvectors of $\mathbf{A}$ corresponding to the same eigenvalue λ, then any linear combination of $\mathbf{u}$ and $\mathbf{v}$ is also an eigenvalue

$$\mathbf{A}(\alpha\mathbf{u} + \beta\mathbf{v}) = \alpha\mathbf{A}\mathbf{u} + \beta\mathbf{A}\mathbf{v} = \alpha\lambda\mathbf{u} + \beta\lambda\mathbf{v} = \lambda(\alpha\mathbf{u} + \beta\mathbf{v}).$$

So the set of all eigenvectors corresponding to a particular eigenvalue λ is a linear subspace of $\mathbb{R}^n$.

Remark. For a linear system of equations, the matrix $\mathbf{A}$ can have a different number of rows and of columns. To ask about the eigenvalues of such a matrix $\mathbf{A}$ would be meaningless. Eigenvalues and eigenvectors make sense only when the vectors $\mathbf{x}$ and $\mathbf{A}\mathbf{x}$ have the same number of rows or when the matrix $\mathbf{A}$ is square.

Remark. Eigenvalues and eigenvectors are useful for algorithmic and physical reasons. Algorithmically, eigenvalue analysis can simplify solutions of certain problems by reducing a coupled systems of equations to a collection of scalar problems. Physically, eigenvalue analysis can give insight into the behavior of evolving systems governed by linear equations. The most familiar examples include the study of *resonance* (of musical instruments when struck or plucked or bowed) and of *stability*.

11.1 How to determine the eigenvalues of a matrix?

Recall the identity matrix $\mathbf{I}$, the $n \times n$ matrix with the property that $\mathbf{Iu} = \mathbf{u}$ for any vector $\mathbf{u} \in \mathbb{R}^n$.

If $\mathbf{Av} = \lambda\mathbf{v}$ with $\mathbf{v} \neq \mathbf{0}$ then we can write this as $\mathbf{Av} = \lambda\mathbf{Iv}$ and rearranging gives

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}. \tag{11.3}$$

So if λ is an eigenvalue of $\mathbf{A}$ the matrix $(\mathbf{A} - \lambda\mathbf{I})$ has a nonzero null vector $\mathbf{v}$. This can happen only if the matrix $(\mathbf{A} - \lambda\mathbf{I})$ is singular.

So one way to find the eigenvalues of a matrix is to try to figure out the values of λ for which $(\mathbf{A} - \lambda\mathbf{I})$ is singular. Note that this matrix is simply the matrix $\mathbf{A}$ with the value λ subtracted from each diagonal element.

We will only consider the case $n = 2$ in detail, so $\mathbf{A}$ is a 2×2 matrix and

$$\mathbf{A} - \lambda\mathbf{I} = \begin{bmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{bmatrix}.$$

For the 2×2 case, we know that this matrix is singular only if its determinant is zero, where

$$\det(\mathbf{A}) = a_{11}a_{22} - a_{12}a_{21}.$$

So we get

$$\begin{aligned} \det(\mathbf{A} - \lambda\mathbf{I}) &= (a_{11} - \lambda)(a_{22} - \lambda) - a_{12}a_{21} \\ \det(\mathbf{A} - \lambda\mathbf{I}) &= \lambda^2 - (a_{11} + a_{22})\lambda + a_{11}a_{22} - a_{12}a_{21} \end{aligned} \tag{11.4}$$

Remember that for a particular matrix we know the a_{ij} values, just some numbers, and we are trying to find λ so that this determinant is zero. This is just a quadratic equation in λ.

Example 220. For the matrix in (11.1), we get

$$\mathbf{A} - \lambda\mathbf{I} = \begin{bmatrix} 3 - \lambda & 0 \\ 0 & -2 - \lambda \end{bmatrix}$$

and so

$$\det(\mathbf{A} - \lambda\mathbf{I}) = -(3 - \lambda)(2 + \lambda).$$

In this case, the quadratic is already factored for us, and we see that this can be zero only if $\lambda = 3$ or $\lambda = -2$. We already knew that $\lambda = 3$ was an eigenvalue

because of (11.2). We now see that there should also be vectors $\mathbf{v}$ for which $\mathbf{A}\mathbf{v} = -2\mathbf{v}$. In fact, there are

$$\mathbf{v} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \Longrightarrow \quad \mathbf{A}\mathbf{v} = \begin{bmatrix} 0 \\ -2 \end{bmatrix} = (-2) \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{11.5}$$

or any multiple of this vector is also an eigenvector.

In the next section, we will see how to determine the eigenvectors once we know the eigenvalues. First, some more examples of computing eigenvalues.

Example 221. Let

$$\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix} \quad \text{and so} \quad \mathbf{A} - \lambda\mathbf{I} = \begin{bmatrix} 3-\lambda & 2 \\ 4 & 1-\lambda \end{bmatrix} \tag{11.6}$$

Then $\det(\mathbf{A} - \lambda\mathbf{I}) = (3-\lambda)(1-\lambda) - 8 = (\lambda - 5)(\lambda + 1)$ so the eigenvalues are $\lambda_1 = 5$ and $\lambda_2 = -1$.

Example 222. Let

$$\mathbf{A} = \begin{bmatrix} 5 & -5 \\ -2 & 2 \end{bmatrix} \quad \text{and so} \quad \mathbf{A} - \lambda\mathbf{I} = \begin{bmatrix} 5-\lambda & -5 \\ -2 & 2-\lambda \end{bmatrix} \tag{11.7}$$

Then $\det(\mathbf{A} - \lambda\mathbf{I}) = (5-\lambda)(2-\lambda) - 10 = \lambda^2 - 7\lambda$. So the eigenvalues are $\lambda_1 = 0$ and $\lambda_2 = 7$. Note that $\mathbf{A}$ has an eigenvalue equal to 0. It means that $\mathbf{A} - 0\mathbf{I}$ is singular, *i.e.* $\mathbf{A}$ is singular.

Any singular matrix has at least one eigenvalue equal to zero, and any null vector of the matrix is an eigenvector associated with this eigenvalue, $\mathbf{A}\mathbf{v} = 0\mathbf{v}$. For the matrix (11.10), the vector $\mathbf{v} = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$ is a null vector of the matrix and hence an eigenvector $\mathbf{A}\mathbf{v} = \mathbf{0} = 0\mathbf{v}$.

For any 2×2 matrix, computing $\det(\mathbf{A} - \lambda\mathbf{I})$ will give a quadratic equation in λ. This quadratic equation can always be factored in the form

$$(\lambda - \lambda_1)(\lambda - \lambda_2) \tag{11.8}$$

where λ_1 and λ_2 are two numbers, though they may be complex numbers! Note that even matrices with real elements can have complex eigenvalues. They often play an important role in analyzing differential equations.

For some matrices, it may happen that $\lambda_1 = \lambda_2$. The eigenvalues are said to be repeated. For example, consider

$$\mathbf{A} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}. \tag{11.9}$$

For this matrix, we have

$$\det(\mathbf{A} - \lambda\mathbf{I}) = (\lambda - 4)^2$$

and so $\lambda_1 = \lambda_2 = 4$. Note that for this matrix any vector is an eigenvector, $\mathbf{A}\mathbf{u} = 4\mathbf{u}$ for every $\mathbf{u} \in \mathbb{R}^2$.

Example 223. Let

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \text{and so} \quad \mathbf{A} - \lambda\mathbf{I} = \begin{bmatrix} -\lambda & 1 \\ -1 & -\lambda \end{bmatrix} \tag{11.10}$$

Then $\det(\mathbf{A} - \lambda\mathbf{I}) = \lambda^2 + 1 = (\lambda - i)(\lambda + i)$. So the eigenvalues are $\lambda_1 = i$ and $\lambda_2 = -i$. The eigenvalues are complex. They are complex conjugates of each other since the matrix $\mathbf{A}$ is real.

If $n > 2$ then it is harder to find the eigenvalues. We still want to find values of λ for which $(\mathbf{A} - \lambda\mathbf{I})$ is singular. For larger matrices, the determinant can also be defined, and the matrix is singular if the determinant is zero. The determinant of $(\mathbf{A} - \lambda\mathbf{I})$ turns out to be a polynomial $p(\lambda)$ of degree n in λ, which can always be factored as

$$p(\lambda) = (\lambda - \lambda_1) \dots (\lambda - \lambda_n) \tag{11.11}$$

and so a matrix of degree n always has exactly n eigenvalues (though some may be repeated). Some may be complex (if $\mathbf{A}$ is real the complex ones always occur in conjugate pairs).

Definition 224. The set of all eigenvalues of a matrix $\mathbf{A}$ is called the *spectrum* of $\mathbf{A}$, a subset of $\mathbb{C}$.

We will not learn how to do this by hand for larger matrices. It is usually very messy just to determine the polynomial, and then there is usually no way to determine the roots except by using a numerical method. Actually numerical methods for determining eigenvalues do not determine this polynomial and search for its roots — it is too difficult to do this way. There are many different algorithms that have been invented for approximating eigenvalues, too complicated to outline here.

There are some cases where it is easy to determine the eigenvalues of a matrix.

- If the matrix is diagonal then the n values on the diagonal are the eigenvalues of the matrix. An eigenvector for the $\lambda_j = a_{jj}$ is given by the j-th column of the identity matrix (the vector with a 1 in the j-th component and 0 everywhere else).
- If the matrix is upper triangular or lower triangular then the eigenvalues are again the diagonal elements, though the eigenvectors are not just the unit vectors in this case.

Finally, we introduce a definition.

Definition 225. The *algebraic multiplicity* for an eigenvalue λ is the multiplicity of λ as a root of the characteristic polynomial.

11.2 How to find the eigenvectors?

Once we know the eigenvalues of a matrix, how do we compute the eigenvectors? Recall that if λ is an eigenvalue then $(\mathbf{A} - \lambda\mathbf{I})$ is singular, and (11.3) just says that an eigenvector $\mathbf{v}$ should be a null vector of this matrix.

So, the set of all eigenvectors for the eigenvalue λ is just the null space of the matrix $(\mathbf{A} - \lambda\mathbf{I})$. This space is called the *eigenspace* of the matrix associated with this eigenvalue.

Example 226. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix}.$$

We know $\lambda_2 = -2$ is an eigenvalue, for example. The associated eigenspace is the null space of

$$\mathbf{A} + 2\mathbf{I} = \begin{bmatrix} 5 & 0 \\ 0 & 0 \end{bmatrix}.$$

The null space of this matrix is span $\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$ and this is the eigenspace.

Example 227. For the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix},$$

the eigenvalues are $\lambda_1 = 5$ and $\lambda_2 = -1$. Let's find the null space for $\lambda_1 = 5$:

$$\mathbf{A} - 5\mathbf{I} = \begin{bmatrix} -2 & 2 \\ 4 & -4 \end{bmatrix}.$$

The second column is -1 times the first, so $\mathbf{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is a null vector, for example, and is an eigenvector of $\mathbf{A}$,

$$\mathbf{A}\mathbf{v} = \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} = 5\mathbf{v}.$$

Example 228. For the matrix

$$\mathbf{A} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix},$$

$\lambda_1 = \lambda_2 = 4$ and

$$\mathbf{A} - 4\mathbf{I} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The null space of this matrix is all of $\mathbb{R}^2$, so every vector is an eigenvector, as we already knew. In this case the eigenspace is 2-dimensional. This can only happen for repeated roots. Isolated roots have 1-dimensional eigenspaces, even for large matrices.

As stated previously, the set of eigenvectors for an eigenvalue forms a subspace of $\mathbb{R}^n$. It is known as the eigenspace. The dimension of the eigenspace is the number of linearly independent eigenvectors.

Definition 229. The *geometric multiplicity* for an eigenvalue λ is the dimension for the eigenspace associated with the eigenvalue λ.

So far, we have seen examples where the algebraic multiplicity and the geometric multiplicity are equal. In general, this is not true. The algebraic multiplicity is always greater or equal to the geometric multiplicity.

Example 230. Consider the matrices

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix}.$$

Both matrices have characteristic polynomial $(\lambda - 2)^3$. 2 has an algebraic multiplicity equal to 3. For the matrix $\mathbf{A}$, we can choose three independent eigenvectors

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

So the geometric multiplicity is also 3. However, for the matrix $\mathbf{B}$, we can find only one independent eigenvector

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

So the geometric multiplicity is only 1.

11.3 Eigenvalue decomposition

An eigenvalue decomposition of matrix $\mathbf{A}$ is a factorization

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}. \tag{11.12}$$

Here the matrix $\mathbf{V}$ is nonsingular and $\mathbf{\Lambda}$ is a diagonal matrix. Note that such a decomposition does not always exist.

The decomposition (11.12) can be rewritten

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \tag{11.13}$$

or

$$\mathbf{A}\left[\begin{array}{c|c|c} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{array}\right] = \left[\begin{array}{c|c|c} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{array}\right] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}. \tag{11.14}$$

This implies that $\mathbf{A}\mathbf{v}_j = \lambda_j \mathbf{v}_j$ and that the j-th column of $\mathbf{V}$ is an eigenvector of $\mathbf{A}$ associated with the eigenvalue λ_j.

The eigenvalue decomposition (11.12) expresses a change of basis to eigenvector coordinates. If $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, we have

$$\mathbf{V}^{-1}\mathbf{b} = \mathbf{\Lambda}\mathbf{V}^{-1}\mathbf{x}.$$

To compute $\mathbf{A}\mathbf{x}$, we can expand $\mathbf{x}$ in the basis of columns of $\mathbf{V}$, apply $\mathbf{\Lambda}$, and interpret the result as a vector of coefficients for a linear combination of the columns of $\mathbf{V}$. To solve $\mathbf{A}\mathbf{x} = \mathbf{b}$, we can expand $\mathbf{b}$ in the basis of columns of $\mathbf{V}$, solve a diagonal system with $\mathbf{\Lambda}$, and interpret the result as a vector of coefficients for a linear combination of the columns of $\mathbf{V}$.

Definition 231. An $n \times n$ matrix $\mathbf{A}$ is diagonalizable when it has an eigenvalue decomposition $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$.

Note that, from the eigendecomposition (11.12), we can not conclude that the matrix $\mathbf{A}$ behaves like the diagonal matrix $\mathbf{\Lambda}$. The statement depends on what aspect of behavior one measures and on the matrix $\mathbf{V}$. A great deal of information can be contained in the matrix $\mathbf{V}$. For example, the determinant of $\mathbf{A}$ satisfies

$$\det(\mathbf{A}) = \det(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}) = \det(\mathbf{V})\det(\mathbf{\Lambda})\det(\mathbf{V})^{-1} = \det(\mathbf{\Lambda}) = \prod_{i=1}^{n} \lambda_i \quad (11.15)$$

and the trace of $\mathbf{A}$ satisfies

$$\operatorname{tr}(\mathbf{A}) = \operatorname{tr}(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}) = \operatorname{tr}(\mathbf{V}^{-1}\mathbf{V}\mathbf{\Lambda}) = \operatorname{tr}(\mathbf{\Lambda}) = \sum_{i=1}^{n} \lambda_i. \quad (11.16)$$

However, the 2-norm of $\mathbf{A}$ satisfies

$$\|\mathbf{A}\|_2 = \|\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}\|_2 \leq \|\mathbf{V}\|_2 \|\mathbf{\Lambda}\|_2 \|\mathbf{V}^{-1}\|_2 = \kappa_2(\mathbf{V})\|\mathbf{\Lambda}\|_2. \quad (11.17)$$

There are matrices for which this bound is sharp.

It sometimes happens that the matrix $\mathbf{A}$ is diagonalizable and that the n linearly independent eigenvectors can be chosen orthogonal. In this case, the matrix $\mathbf{A}$ is diagonalizable and there exists an orthogonal matrix $\mathbf{Q}$ such that

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T. \quad (11.18)$$

Theorem 232. *A symmetric matrix $\mathbf{A}$ is diagonalizable. Its eigenvalues are real and there exists an orthogonal matrix $\mathbf{Q}$ such that*

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \quad \textit{where} \quad \mathbf{Q}^T\mathbf{Q} = \mathbf{I}. \quad (11.19)$$

One final factorization is actually the one that is most useful in numerical analysis because all matrices can be factored in this way.

Theorem 233. *Every square matrix* $\mathbf{A}$ *has a Schur factorization*

$$\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^* \tag{11.20}$$

where $\mathbf{Q}$ *is unitary and* $\mathbf{T}$ *is upper triangular.*

Note that the triangular matrix $\mathbf{T}$ can have complex entries. Matrices $\mathbf{A}$ and $\mathbf{T}$ have the same eigenvalues. So the eigenvalues of $\mathbf{A}$ necessarily appear on the diagonal of $\mathbf{T}$. Alternatively, for a real square matrix, we can write a real reduction to an upper quasi-triangular.

Theorem 234. *Every square real matrix* $\mathbf{A}$ *has a real Schur factorization:*

$$\mathbf{A} = \mathbf{Q}\begin{bmatrix} R_{11} & R_{12} & \dots & R_{1m} \\ 0 & R_{22} & & R_{2m} \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & R_{mm} \end{bmatrix}\mathbf{Q}^T \tag{11.21}$$

where $\mathbf{Q}$ *is orthogonal and each block* R_{ii} *is either a* 1×1 *matrix or a* 2×2 *matrix having complex conjugate eigenvalues.*

Example 235. Consider a matrix $\mathbf{B} \in \mathbb{R}^{m\times n}$. Recall that the Frobenius norm is

$$\|\mathbf{B}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}^2} = \sqrt{\mathrm{tr}\,(\mathbf{B}^T\mathbf{B})}.$$

The matrix $\mathbf{B}^T\mathbf{B}$ is symmetric. So it is diagonalizable, *i.e.* $\mathbf{B}^T\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. So we get

$$\|\mathbf{B}\|_F = \sqrt{\mathrm{tr}\,(\mathbf{B}^T\mathbf{B})} = \sqrt{\mathrm{tr}\,(\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T)} = \sqrt{\mathrm{tr}\,(\mathbf{\Lambda}\mathbf{Q}^T\mathbf{Q})}.$$

Since $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, we get

$$\|\mathbf{B}\|_F = \sqrt{\mathrm{tr}\,(\mathbf{\Lambda})} = \sqrt{\sum_{i=1}^{n}\lambda_i(\mathbf{B}^T\mathbf{B})}. \tag{11.22}$$

Example 236. Consider a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ which is full rank. The matrix $\mathbf{A}^T\mathbf{A}$ is symmetric. So it is diagonalizable, *i.e.* $\mathbf{A}^T\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. Note that the eigenvalues of $\mathbf{A}^T\mathbf{A}$ are always positive. If we have $\mathbf{A}^T\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$, then we get $\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = \lambda\mathbf{x}^T\mathbf{x}$ and

$$\|\mathbf{A}\mathbf{x}\|_2^2 = \lambda\|\mathbf{x}\|_2^2$$

which implies that $\lambda \geq 0$. The matrix $\mathbf{A}$ is full rank which implies that $\mathbf{A}^T\mathbf{A}$ is invertible. Consequently, all the eigenvalues λ_i are strictly positive. Denote σ_i the positive square root of λ_i and $\mathbf{\Sigma}$ the square diagonal matrix whose entries are $\sigma_1, \dots, \sigma_n$. We have $\mathbf{A}^T\mathbf{A} = \mathbf{Q}\mathbf{\Sigma}^2\mathbf{Q}^T$. We will show that $\mathbf{\Sigma}$ and $\mathbf{Q}$ are two components of the singular value decomposition of $\mathbf{A}$. Consider the matrix

$\mathbf{U} = \mathbf{A}\mathbf{Q}\mathbf{\Sigma}^{-1}$ which belongs to $\mathbb{R}^{m\times n}$. The matrix $\mathbf{U}$ has orthonormal column vectors. Indeed, we have

$$\mathbf{U}^T\mathbf{U} = \mathbf{\Sigma}^{-1}\mathbf{Q}^T\mathbf{A}^T\mathbf{A}\mathbf{Q}\mathbf{\Sigma}^{-1} = \mathbf{\Sigma}^{-1}\mathbf{Q}^T\mathbf{Q}\mathbf{\Sigma}^2\mathbf{Q}^T\mathbf{Q}\mathbf{\Sigma}^{-1} = \mathbf{I}.$$

Finally, we write

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

where $\mathbf{U}$ is an $m \times n$ matrix with orthonormal column vectors, $\mathbf{\Sigma}$ is an $n \times n$ diagonal matrix whose entries are positive, and $\mathbf{V} = \mathbf{Q}$ is an $n \times n$ orthogonal matrix. The scalar σ_i are called the singular values of $\mathbf{A}$. The column vectors of $\mathbf{V}$ are called the right singular vectors. The column vectors of $\mathbf{U}$ are called the left singular vectors. Note that we have

$$\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i. \tag{11.23}$$

11.4 Case of symmetric matrices

In this section, we simplify matters by considering only matrices that are real and symmetric. Recall that such a matrix has real eigenvalues and a complete set of orthonormal eigenvectors. We denote $\lambda_1, \ldots, \lambda_n$ the set of eigenvalues ordered in a non-decreasing way and $\mathbf{q}_1, \ldots, \mathbf{q}_n$ the associated orthonormal eigenvectors.

11.4.1 Rayleigh quotient

Consider a real symmetric matrix $\mathbf{A}$. The Rayleigh quotient of a non-zero vector $\mathbf{x} \in \mathbb{R}^n$ is the scalar

$$\rho(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{A}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}. \tag{11.24}$$

Notice that if $\mathbf{x}$ is an eigenvector, then its Rayleigh quotient is equal to the associated eigenvalue. The Rayleigh quotient plays a big role in the computation of eigenvalues and eigenvectors. It has some nice properties.

- The Rayleigh quotient is bounded and ranges over the interval $[\lambda_1, \lambda_n]$, *i.e.* for any nonzero vector $\mathbf{x}$, we have

$$\lambda_1 \leq \rho(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{A}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \leq \lambda_n. \tag{11.25}$$

- The decomposition is

$$\mathbf{A}\mathbf{x} = \rho(\mathbf{x})\mathbf{x} + (\mathbf{A}\mathbf{x} - \rho(\mathbf{x})\mathbf{x})$$

  orthogonal, *i.e.*

$$\mathbf{x}^T\left(\mathbf{A}\mathbf{x} - \rho(\mathbf{x})\mathbf{x}\right) = 0.$$

- The Rayleigh quotient is the scalar minimizing the norm $\|\mathbf{A}\mathbf{x} - \alpha\mathbf{x}\|_2$, *i.e.*

$$\|\mathbf{A}\mathbf{x} - \rho(\mathbf{x})\mathbf{x}\|_2 \leq \|\mathbf{A}\mathbf{x} - \alpha\mathbf{x}\|_2, \quad \forall\alpha \in \mathbb{R}.$$

- The eigenvectors of $\mathbf{A}$ are the stationary points for the Rayleigh quotient. Indeed, we have

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

The Rayleigh quotient appears in the definition of the matrix 2-norm. Recall that the matrix 2-norm is associated with the Euclidean vector norm, namely

$$\|\mathbf{B}\|_2 = \max_{\mathbf{x}\neq 0} \frac{\|\mathbf{B}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \max_{\mathbf{x}\neq 0} \sqrt{\frac{(\mathbf{B}\mathbf{x})^T\mathbf{B}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}} = \max_{\mathbf{x}\neq 0} \sqrt{\frac{\mathbf{x}^T\mathbf{B}^T\mathbf{B}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}}. \tag{11.26}$$

The matrix $\mathbf{B}^T\mathbf{B}$ is symmetric. So the matrix 2-norm of $\mathbf{B}$ is the square root of the largest value for the Rayleigh quotient of $\mathbf{B}^T\mathbf{B}$. Based on (11.25), the matrix 2-norm of $\mathbf{B}$ is equal to the square root of the largest eigenvalue of of $\mathbf{B}^T\mathbf{B}$,

$$\|\mathbf{B}\|_2 = \sqrt{\lambda_{\max}(\mathbf{B}^T\mathbf{B})}. \tag{11.27}$$

Note that the eigenvalues of $\mathbf{B}^T\mathbf{B}$ are always positive. Indeed we have

$$\text{If } \mathbf{B}^T\mathbf{B}\mathbf{x} = \lambda\mathbf{x}, \text{ then } \mathbf{x}^T\mathbf{B}^T\mathbf{B}\mathbf{x} = \lambda\mathbf{x}^T\mathbf{x} \quad \Rightarrow \quad \|\mathbf{B}\mathbf{x}\|_2^2 = \lambda\|\mathbf{x}\|_2^2$$

which implies that $\lambda \geq 0$.

11.4.2 Power iteration

Consider the following algorithm. Note that we did not specify any stopping

Algorithm 11 Power iteration

Set $\mathbf{v}^{(0)}$ an arbitrary vector in $\mathbb{R}^n$ of norm 1, $\|\mathbf{v}^{(0)}\|_2 = 1$.
for $k = 1, 2, \ldots$ **do**
  $\mathbf{w} = \mathbf{A}\mathbf{v}^{(k-1)}$
  $\mathbf{v}^{(k)} = \mathbf{w}/\|\mathbf{w}\|_2$
  $\lambda^{(k)} = \left(\mathbf{v}^{(k)}\right)^T \mathbf{A}\mathbf{v}^{(k)}$
end for

criterion for this algorithm. In practice, termination criteria are very important.

To analyze the power iteration algorithm, notice that $\mathbf{v}^{(k)}$ is proportional to the vector $\mathbf{A}^k\mathbf{v}^{(0)}$. Next, we write $\mathbf{v}^{(0)}$ as a linear combination of the orthonormal eigenvectors

$$\mathbf{v}^{(0)} = a_1\mathbf{q}_1 + \ldots + a_n\mathbf{q}_n.$$

Recall that $\mathbf{A}\mathbf{q}_i = \lambda_i\mathbf{q}_i$ and $\mathbf{A}^k\mathbf{q}_i = \lambda_i^k\mathbf{q}_i$. So we get

$$\mathbf{v}^{(k)} = c_k\left(\lambda_1^k a_1\mathbf{q}_1 + \ldots + \lambda_n^k a_n\mathbf{q}_n\right)$$

where c_k is a scalar for the normalization of $\mathbf{v}^{(k)}$

$$c_k = \frac{1}{\|\lambda_1^k a_1\mathbf{q}_1 + \ldots + \lambda_n^k a_n\mathbf{q}_n\|_2} = \frac{1}{\sqrt{\lambda_1^{2k}a_1^2 + \ldots + \lambda_n^{2k}a_n^2}}.$$

Factorizing by λ_n^k, we obtain

$$\mathbf{v}^{(k)} = \frac{1}{\sqrt{(\lambda_1/\lambda_n)^{2k} a_1^2 + \ldots + a_n^2}} \left[\left(\frac{\lambda_1}{\lambda_n}\right)^k a_1 \mathbf{q}_1 + \left(\frac{\lambda_2}{\lambda_n}\right)^k a_2 \mathbf{q}_2 + \ldots + a_n \mathbf{q}_n \right].$$

From here, we get the following result.

Theorem 237. *Suppose the eigenvalues of* $\mathbf{A}$ *satisfy* $|\lambda_n| > |\lambda_{n-1}| \geq \ldots \geq |\lambda_1|$ *and* $\mathbf{q}_n^T \mathbf{v}^{(0)} = a_n \neq 0$. *Then the iterates for the power iteration algorithm satisfy*

$$\|\mathbf{v}^{(k)} - \mathbf{q}_n\|_2 = \mathcal{O}\left(|\frac{\lambda_{n-1}}{\lambda_n}|^k\right), \quad |\lambda^{(k)} - \lambda_n| = \mathcal{O}\left(|\frac{\lambda_{n-1}}{\lambda_n}|^{2k}\right) \tag{11.28}$$

when $k \to +\infty$.

Depending on the normalization step, the iterates can actually converge to $-\mathbf{q}_n$. But the convergence rates do not change. Notice that the approximate eigenvalue converges at a rate twice faster than the vector iterates. However, power iteration is of limited use. First, it can find only the eigenvector corresponding to the largest eigenvalue. Second, the convergence is linear, reducing the error by a constant factor $\approx |\lambda_{n-1}/\lambda_n|$. The convergence can be slow when the two eigenvalues are close to each other. Combined with other ingredients, this tool can result in a powerful algorithm.

11.5 Analyzing matrix powers with the eigenvalue decomposition

In this section we discuss how to analyze the behavior of a matrix raised to a large power using the eigenvalue decomposition. Suppose $\mathbf{A} \in \mathbb{R}^{n \times n}$ is diagonalizable, so its eigenvalue decomposition is

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}.$$

In many situations we encounter a matrix raised to a power, $\mathbf{A}^k$, for some large integer k. Substituting in the eigenvalue decomposition of A gives

$$\begin{aligned}
\mathbf{A}^k &= (\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1})^k \\
&= (\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1})(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}) \cdots (\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1})(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}) \\
&= \mathbf{V}\mathbf{\Lambda}(\mathbf{V}^{-1}\mathbf{V})\mathbf{\Lambda}\mathbf{V}^{-1} \cdots \mathbf{V}\mathbf{\Lambda}(\mathbf{V}^{-1}\mathbf{V})\mathbf{\Lambda}\mathbf{V}^{-1} \\
&= \mathbf{V}\mathbf{\Lambda}\mathbf{I} \cdots \mathbf{I}\mathbf{\Lambda}\mathbf{V}^{-1} \\
&= \mathbf{V}\mathbf{\Lambda}\mathbf{\Lambda} \cdots \mathbf{\Lambda}\mathbf{V}^{-1} \\
&= \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{-1}.
\end{aligned}$$

Since $\mathbf{\Lambda}$ is diagonal, $\mathbf{\Lambda}^k$ is easy to compute:

$$\mathbf{\Lambda}^k = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}^k = \begin{bmatrix} \lambda_1^k & & & \\ & \lambda_2^k & & \\ & & \ddots & \\ & & & \lambda_n^k \end{bmatrix}.$$

Definition 238. The *spectral radius* of a square matrix $\mathbf{A}$ is the maximum of the absolute values of the eigenvalues of $\mathbf{A}$. We denote the spectral radius by

$\rho(\mathbf{A} = \max_{1 \leq i \leq n} |\lambda_i|$

where λ_i is the$i$-th eigenvalue of $\mathbf{A}$.

If $\rho(\mathbf{A}) < 1$ then $\lambda_i^k \to 0$ as $k \to \infty$ for each $i = 1, 2, \ldots, n$. In this case $\mathbf{\Lambda}^k$tends to the zero matrix as $k \to \infty$, which implies that

$\mathbf{A}^k = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{-1} \to \mathbf{V0V}^{-1} = \mathbf{0}$

as $k \to \infty$. If $\rho(\mathbf{A}) > 1$, then $\mathbf{A}$, has at least one eigenvalue with magnitude larger than 1. Without loss of generality, assume $|\lambda_1| > 1$. Then $|\lambda_1|^k \to \infty$ as $k \to \infty$. This means that at least one entry of$\mathbf{\Lambda}^k$ will grow arbitrarily large in magnitude as k increases, which will cause entries of $\mathbf{A}^k$ to grow without bound for large k. If $\rho(\mathbf{A}) = 1$ then, by a similar analysis, at least one of the entries of $\mathbf{\Lambda}^k$ will always have magnitude 1 for any value of k and so the entries in $\mathbf{A}^k$ will neither blow up nor decay to zero.

11.6 Useful commands in MATLAB

Here are a few commands in MATLAB useful for this section.

- `null(A)` computes an orthonormal basis for the null space of $\mathbf{A}$, when it exists.
- `d = eig(A)` returns a vector of the eigenvalues of matrix $\mathbf{A}$.
- `[V, D] = eig(A)` produces matrices of eigenvalues $\mathbf{D}$ and eigenvectors $\mathbf{V}$ of matrix A, so that $\mathbf{AV} = \mathbf{VD}$.
- `T = schur(A)` returns the Schur matrix $\mathbf{T}$.
- `T = schur(A,'real')` returns the matrix $\mathbf{T}$ for the real Schur factorization.
- `[U, T] = schur(A)` produces the Schur matrix $\mathbf{T}$ and a unitary matrix $\mathbf{U}$, so that $\mathbf{AU} = \mathbf{UT}$ and $\mathbf{U}^*\mathbf{U} = \mathbf{I}$.
- `[Q, T] = schur(A,'real')` produces the real Schur factorization: a matrix $\mathbf{T}$ and an orthogonal matrix $\mathbf{Q}$, so that $\mathbf{AQ} = \mathbf{QT}$ and $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$.

12 Iterative solvers

We return to the problem of solving the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$. The techniques for solving this problem we have seen so far (e.g. Gaussian elimination via the LU factorization) are called *direct* solvers/methods since, given infinite numerical precision, they directly manipulate the given entries of $\mathbf{A}$ and $\mathbf{b}$ to obtain the exact entries of the solution vector, $\mathbf{x}$. These methods find the exact solution in a \textbf{finite} number of steps. Alternatively, there are *iterative* solvers/methods which, given some initial guess at the solution, $\mathbf{x}^{(0)}$, compute successively better and better approximations $\mathbf{x}^{(k)}$ to the exact solution, denoted $\mathbf{x}^*$. Given enough iterations these methods can often guarantee that their approximations become arbitrarily close to the true solution $\mathbf{x}^*$, however they typically do not produce it exactly.

12.1 Sparse matrices

Definition 239. A **sparse** matrix is one in which most of the entries are zero. A matrix which has mostly nonzero entries is called **dense**.

Note that a matrix with 10001 zero entries and 9999 zero entries would technically be considered sparse by this definition. However, we will mostly be interested in matrices that are very sparse, e.g. ones with, say, 10% or fewer nonzero entries.

Example 240. For large values of n, diagonal $n \times n$ matrices are considered to be sparse since n of their entries are nonzero and $n^2 - n$ of their entries are 0.

Example 241. The following matrix is used when numerically solving a second order ordinary differential equation. This matrix is sparse. In fact, in obtaining numerical solutions to differential equations, many of the matrices one deals with are sparse.

$$\begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \dots & 0 & 1 & -2 \end{bmatrix}$$

When one works with very large matrices, the amount of memory an algorithm requires can become a concern. In practice situations often occur where the matrix itself, specifying the system to be solved, is too large to fit into memory (RAM) all at once. When a matrix is sparse, it can be stored in a number of special formats which ignore the 0 entries and only keep track of the nonzero ones. This can greatly reduce the amount of memory needed to store the matrix and reduces the cost of matrix-vector multiplications involving this matrix. If an $n \times n$ sparse matrix has at most M entries in each row and is stored in a sparse format, then a matrix vector multiplication costs only $\mathcal{O}(Mn) = \mathcal{O}(n)$

floating point operations (at most $\mathcal{O}(M)$ along each row, with n total rows). Recall that for a dense $n \times n$ matrix, this operation would cost $\mathcal{O}(n^2)$ floating point operations.

When should one use iterative solvers?

Iterative methods typically involve performing many matrix-vector products rather than manipulations of the matrix itself (as in LU factorization). This consideration, along with the ones in the preceeding section give us an idea of when it is reasonable to employ an iterative solver instead of a direct one:

1. When the system matrix, $\mathbf{A}$, is large and sparse. If $\mathbf{A}$ is large and sparse then the fact that matrix-vector multiplications are cheaper for sparse matrices can bring the computational complexities of iterative methods far below those of direct ones.

2. When $\mathbf{A}$ is large enough that memory becomes an issue. This is because one can perform matrix-vector multiplications without loading the entire matrix into memory. For concreteness, suppose we wanted to compute the product $\mathbf{z} = \mathbf{A}\mathbf{w}$. Instead of performing the multiplication all at once, we could find each of the entries of $\mathbf{z}$ individually. If a_{ij} is the (i,j) entry of $\mathbf{A}$ and w_i the i-th entry of $\mathbf{w}$, then the i-th entry of the product-vector $\mathbf{z}$ is given by

$$\mathbf{z}_i = \sum_{j=1}^{n} a_{ij} w_j.$$

   Notice that this computation involves only the ith row of $\mathbf{A}$. Therefore, we can find $\mathbf{z}$ entry-by-entry by loading individual rows of $\mathbf{A}$ one at a time and taking their dot product with $\mathbf{w}$. This series of operations is much less memory intensive than the standard way of computing a matrix-vector product. Since matrix-vector products can be made memory-friendly and iterative methods rely on matrix-vector products, iterative methods can be useful when $\mathbf{A}$ is extremely large.

12.2 Simple Iteration

Again, consider the problem of solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for some given $\mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{b} \in \mathbb{R}^n$, where $\mathbf{A}$ is invertible. Simple iteration is an iterative algorithm for solving this problem approximately for $\mathbf{x}$:

Algorithm 12 Simple iteration

Given $\mathbf{x}^{(0)} \in \mathbb{R}^n$,
for $k = 1, 2, 3, \ldots$ **do**
  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)})$
end for

At each step, a new iterate ($\mathbf{x}^{(k+1)}$) is produced by taking the previous iterate ($\mathbf{x}^{(k)}$) and adding to it a matrix multiplied by the residual from the previous iterate ($\mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)})$). Different choices of the matrix $\mathbf{M}$ yield different methods:

- If $\mathbf{M}$ is the diagonal matrix consisting of the diagonal entries of $\mathbf{A}$, then Algorithm 12 is called **Jacobi iteration**.
- If $\mathbf{M}$ is lower triangular and consists of the lower triangular part of $\mathbf{A}$, then Algorithm 12 is called **Gauss-Seidel iteration**.
- if $\mathbf{M}$ is $\omega^{-1}\mathbf{D} - \mathbf{L}$, where $\mathbf{D}$ is the diagonal part of $\mathbf{A}$, $\mathbf{L}$ is the lower triangular part of $\mathbf{A}$, and ω is a relaxation parameter, then Algorithm 12 is called **Successive Over-relaxation** (SOR).

Note that we did not give a stopping condition. One popular choice is to first set some desired tolerance, $\epsilon > 0$, then to terminate the algorithm when two subsequent iterates are within ϵ of each other in norm: $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| \leq \epsilon$. Another is to choose a tolerance, $\epsilon > 0$ and halt the algorithm when the residual of the approximation becomes small enough: $\|\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}\| \leq \epsilon$.

The convergence behavior of simple iteration is controlled by the eigenvalues of a specific matrix. Let

$$\mathbf{e}^{(k)} = \mathbf{A}^{-1}\mathbf{b} - \mathbf{x}^{(k)}$$

denote the error in the k-th iterate of simple iteration. Note that $\mathbf{A}^{-1}\mathbf{b}$ is the exact solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$, so $\mathbf{e}^{(k)}$ is just the difference between the exact solution and $\mathbf{x}^{(k)}$. We can use the definition of $\mathbf{x}^{(k)}$ to rewrite $\mathbf{e}^{(k)}$ as follows

$$\begin{aligned}
\mathbf{e}^{(k)} = \mathbf{A}^{-1}\mathbf{b} - \mathbf{x}^{(k)} &= \mathbf{A}^{-1}\mathbf{b} - \left(\mathbf{x}^{(k-1)} + \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k-1)})\right) \\
&= \mathbf{A}^{-1}\mathbf{b} - \mathbf{x}^{(k-1)} - \mathbf{M}^{-1}\left(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k-1)}\right) \\
&= \left(\mathbf{A}^{-1}\mathbf{b} - \mathbf{x}^{(k-1)}\right) - \mathbf{M}^{-1}\mathbf{A}\left(\mathbf{A}^{-1}\mathbf{b} - \mathbf{x}^{(k-1)}\right) \\
&= \mathbf{e}^{(k-1)} - \mathbf{M}^{-1}\mathbf{A}\mathbf{e}^{(k-1)} \\
&= \left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\right)\mathbf{e}^{(k-1)}.
\end{aligned}$$

The takeaway from this string of equalities is that we may express the error at step k, $\mathbf{e}^{(k)}$, as a matrix times the error at the previous step:

$$\mathbf{e}^{(k)} = \left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\right)\mathbf{e}^{(k-1)}. \tag{12.1}$$

Applying (12.1) to itself recursively gives

$$
\begin{aligned}
\mathbf{e}^{(k)} &= \left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\right)\mathbf{e}^{(k-1)} \\
&= \left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\right)\left(\left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\right)\mathbf{e}^{(k-2)}\right) \\
&= \left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\right)^2\mathbf{e}^{(k-2)} \\
&= \left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\right)^3\mathbf{e}^{(k-3)} \\
&\quad\vdots \\
&= \left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\right)^k\mathbf{e}^{(0)}
\end{aligned}
$$

Hence, the eigenvalues of $\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}$ dictate the convergence behavior of the method. If $\rho(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}) < 1$ then $\mathbf{e}^{(k)} \to \mathbf{0}$ as $k \to \infty$. If $\rho(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}) \geq 1$ then simple iteration will not converge. That $\rho(\mathbf{I}-\mathbf{M}^{-1}\mathbf{A}) < 1$ is a sufficient and necessary condition for convergence. Simple iteration will converge if and only if this condition is met. However, this can be a difficult property to check directly, so we present sufficient (but not necessary) conditions to ensure convergence for the three versions of simple iteration given above.

- **Jacobi iteration**: $\mathbf{A}$ is strictly diagonally dominant. This means that for each row of $\mathbf{A}$, the absolute value of the diagonal entry is larger than the sum of the absolute values of the off-diagonal entries:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|.$$

- **Gauss-Seidel**: $\mathbf{A}$ is either symmetric positive definite or strictly diagonally dominant. A matrix is symmetric positive definite if it is symmetric and for all $\mathbf{u} \neq \mathbf{0}$, $\mathbf{u}^T\mathbf{A}\mathbf{u} > 0$ (equivalently, $\mathbf{A}$ is symmetric positive definite if it is symmetric and has all positive eigenvalues).

- **SOR**: If $\omega\mathbf{M}$ is nonsingular, $\mathbf{A}$ is symmetric positive definite, and $0 < \omega < 2$, then SOR converges.

It should be noted that in practice one usually does not use simple iteration to solve linear systems because there are much better iterative methods available. However, simple iteration is sometimes used as a component of more complicated solvers (e.g. in some multigrid methods).

12.3 Other iterative methods

There are many other iterative solvers which are much more sophisticated. Below we list a few and specify the types of problems for which they are appropriate.

- **Conjugate Gradient**: Works for symmetric positive definite matrices.

- **MINRES**: Works for Hermitian matrices.
- **GMRES**: Works for general matrices.
- **Multigrid**: Typically used for numerical solution of elliptic partial differential equations.

A Notations

- $\mathbb{R}$ is the set of all real numbers (-1, π, $-\sqrt{2}$, ...).
- $\mathbb{C}$ is the set of all complex numbers.
- $a \in \mathbb{R}$ means "a is in $\mathbb{R}$" or "a belongs to $\mathbb{R}$".
- $[a, b]$ denotes an interval: if $a \leq x \leq b$, then $x \in [a, b]$
- (a, b) denotes an interval: if $a < x < b$, then $x \in (a, b)$
- $[a, b)$ denotes an interval: if $a \leq x < b$, then $x \in [a, b)$
- $[a, b] \subset \mathbb{R}$ means that $[a, b]$ is included in $\mathbb{R}$.
- $\mathbb{R} \supset [a, b]$ means that $\mathbb{R}$ contains $[a, b]$.
- $\forall$ means "for all".
- $\exists$ means "there exists".

B Introduction to MATLAB

MATLAB is a very convenient language for testing simple algorithms, analyzing data interactively, and plotting results. This language is widely used by mathematicians and engineers.

Following are some hints on using MATLAB that may be particularly useful for getting started in this class. Check out the MATLAB tutorials linked from the class webpage for more hints.

- To start MATLAB on the PCs in the AS lab, click on the Start menu and then Programs and then on Matlab. MATLAB has a nice user interface that you should experiment with and use to learn effectively.
- Type "`helpwin`" at the MATLAB prompt to bring up some introductory information and browse all the help pages. You might also try the `demo` command in MATLAB.
- Use the `help` (or `helpwin`) and `lookfor` commands in MATLAB to learn about how various commands and functions work. For example,
  `helpwin plot`
  will give you information about how to use the `plot` command,
  `lookfor axis`
  will give a list of all commands whose helpfile contains the string "axis" and might help you find the command you are looking for regarding setting axes on a plot. The `which` command is useful if there may be multiple ".m" files with the same name and you are not sure which is being used.
- Temporary file storage is in the `C:\Temp` directory. This should be where MATLAB works by default (for example if you create a new ".m" file – also called m-file – from MATLAB). When you log out, the `C:\Temp` directory will automatically be moved to `C:\oldtemp` so the next user will have an empty `Temp` folder. If you forget to save something on a flash drive or another account that you need, you can log back on and recover it from `C:\oldtemp`. However, the next time you or someone else logs out, `C:\oldtemp` is overwritten and the previous version is lost forever.

If you compute something in MATLAB, say

```
>> x = sqrt(17)
```

MATLAB will do the computation and then print out the result on the screen. If you want to do the computation but not print out the result, end the command with a semi-colon,

```
>> x = sqrt(17);
```

This is particularly useful if you are doing computations on vectors or matrices where the result may consist of hundreds or thousands of numbers.

If you just type

```
>> sqrt(17)
```

without assigning this to a variable, it will compute it and assign it to a variable named `ans`. You can then use `ans` in a future computation. (But `ans` will be redefined next time you compute something without assigning it elsewhere!)

You can have more than one command on a line if you separate them by semicolons, or by commas if you want the result printed, for example

```
>> a = sqrt(17); b = 12, c = 3
```

the values of `a, b, c` will all be computed and the values of `b` and `c` will be printed out.

In m-file scripts containing lots of commands, it is generally best to put only one command on each line for readability. However, the examples in these notes will sometimes put multiple command on a line to save space.

MATLAB views all numerical variables as matrices. A single number, e.g. `x = 2` can be thought of as a 1×1 matrix. A row vector, e.g.,

```
x = 1:4   which is the same as x = [1 2 3 4] or x = [1, 2, 3, 4]
```

is a 1×4 matrix. A column vector, e.g.

```
x = (1:4)'   which is the same as x = [1; 2; 3; 4]
```

is a 4×1 matrix. The command `size(x)` will tell you the size.

When using vectors only as a data structure to store a collection of numbers, it often doesn't matter whether they are thought of as row vectors or column vectors. In linear algebra it is often important to distinguish between the two. Our vectors will always be column vectors unless it is clear that a row vector is meant. Row vectors can be transformed into column vectors and vice versa by the *transpose operator*, denoted by a superscript T. In MATLAB, the prime symbol is used, so for example `[1; 2; 3]' = [1, 2, 3]`.

In linear algebra we frequently use matrix-vector and matrix-matrix multiplication, which can be easily done in MATLAB. For example, if `A` is a 3×4 matrix and `x` is a 4×1 vector, then the command

```
>>b = A*x
```

computes the matrix-vector product (a 3×1 vector). For the multiplication operator `*` to work, the matrices involved must have appropriate dimensions.

In addition to doing linear algebra, we also frequently use vectors to store the values of a function evaluated at many points. For example, we might set

```
>> x = linspace(-2, 2, 100);
>> y = 2*x + 3;
```

in order to plot a graph of the line $y = 2x + 3$ for $-2 \leq x \leq 2$. The `linspace` command gives a vector of 100 linearly spaced points between -2 and 2. (There's also a command `logspace` that gives logarithmically spaced points.)

Note: if you leave off the semicolon at the end of one of the statements above, MATLAB will print the result, a vector of 100 numbers!

The `linspace` command results in a row vector. If you want `x` to be a column vector you will have to transpose it. For plotting purposes it doesn't matter whether `x` is a row vector or a column vector. The `y` defined above will be either a row or column vector depending on which we have used for `x`. It is possible to plot more than one curve with a single plot command, in which case `Y` could be a matrix and each *column* of the matrix gives the points for one curve. In this case it does matter how it is oriented. For example:

```
>> x = linspace(-2, 2, 100)';
>> Y = [2*x + 3, sin(x)];
>> plot(x,Y)
```

will plot the functions $2x + 3$ and $\sin(x)$ on a single plot. Note that `x` has been defined as a column vector here. Figure B.1 contains the resulting plot

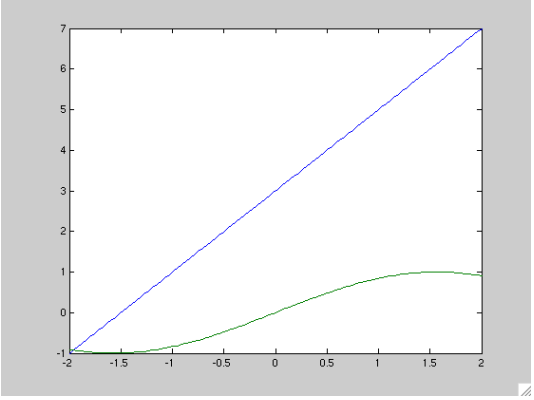


Figure B.1: Plot of $y = 2x + 3$ and $y = \sin(x)$ over $[-2, 2]$

If we want to plot the parabola $y = x^2$ and try to set

```
>> y = x*x;  or  >> y = x^2
```

an error message will result because the vector `x` cannot be multiplied by `x` in the linear algebra sense. What we want is component-wise multiplication, resulting in a new vector with 100 components containing values of x^2 at each point. We can type

```
>> y = x.*x; or >> y = x.^2
```

to obtain this. Figure B.2 contains the resulting plot

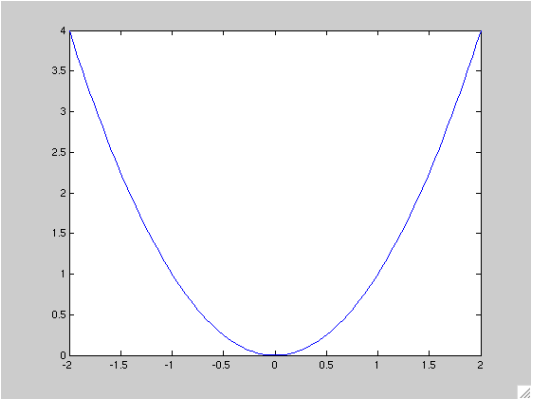


Figure B.2: Plot of $y = x^2$ over $[-2, 2]$

The division operator `/` also has a component-wise version `./`

As another example, performing the above command and then setting

```
>> z = 2*y + y./(5 + x);
```

will set the ith element of `z` to $z_i = 2y_i + y_i/(6 + x_i)$. Note that we can multiply by the scalar 2 using `*` and each element of the vector is multiplied by 2. We can also add the scalar 5 to a vector and each element has 5 added to it.

Operators like `sqrt`, `cos`, `exp` apply to each element of a vector, so setting

```
>> z = exp(x) .* cos(y);
```

sets $z_i = e^{x_i} \cos(y_i)$, provided that x and y have the same size (otherwise an error message results).

The `diary` command is useful for making a record of everything you type and the results that MATLAB computes. If you type

```
>> diary('hw1.txt')
... some other commands ...
>> diary off
```

then all the other commands and responses will be stored in the file `hw1.txt`. This is useful for capturing your homework solutions. You can then edit this file, print it out, and turn it in. (Note: you must say `diary off` before the file can be viewed or printed. You can later say `diary on` to add more to it.)

Note that this file should be in the `C:\Temp` directory. If you are viewing this directory from within the MATLAB window, you may have to select "All Files" in order to see this since by default MATLAB displays only m-files ending with the `.m` extension.

`Script files:` m-files are useful for storing a sequence of commands to be executed. If you have a file called `hw1a.m`, for example, then typing

```
>> hw1a
```

will cause this set of commands to be executed. If you are using the `diary` command to capture things, then you may want to type "`echo on`" before "`hw1a`" so that the commands in `hw1a` are echoed along with MATLAB's response. Otherwise the diary file will only contain the responses, not the commands.

MATLAB has standard control structures such as "for loops" and "while loops". Some examples:

```
x = 4;
for i=1:6
  x = 0.5*(x + 19/x);
end
```

approximates $\sqrt{19}$ using Newton's method. The value of `x` is printed out each time it is modified.

In this loop the vector `1:6` is used to indicate that the loop should be repeated 6 times. Any other vector can be used in the for loop,

```
for n = [1, 4, 9, 16, 25]
  y(n) = sqrt(n);
end
```

will result in `y = [1 2 3 4 5]`.

```
for n = 5:-1:1
  y(n) = n^2;
end
```

will result in `y = [25 16 9 4 1]`.

```
n = 5;
while n > 0
  y(n) = n^2;
  n = n - 1;
end
```

will also result in `y = [25 16 9 4 1]`.

These sorts of loops can be put on a single line at the prompt by using commas. The last loop could be executed by

```
>> n = 5; while n>0, y(n) = n^2; n = n-1; end
```

This is sometimes useful for doing something quick and dirty. However, it is much better to put the loop in a m-file with one command on each line and proper indentation so that it is readable.

If-then-else constructs can also be used,

```
x = rand(1); % a single random number between 0 and 1
if x < 1/3
  y = -1;
elseif x < 2/3
  y = 0;
else
  y = 1;
end
```

will set y to $-1, 0$, or 1 with equal probability. Note that the word "then" does not appear!

The expression `x < 1/3` appearing above evaluates to either 0 (representing False) or 1 (representing True). Logical expressions can be combined using `&` for "and" and `|` for "or",

```
x = rand(1); % a single random number between 0 and 1
if (x < 1/3) |(x == 0.5)
  y = -1;
elseif x < 2/3
  y = 0;
else
  y = 1;
end
```

will set y to 2 if `x < 1/3` or `x == 0.5`. In fact this can be done more simply (but less clearly) by

```
y = 2*(x<1/3 | x==0.5);
```

Note that `==` is used to test for equality, not `=`, which is only used to assign values to variables.

There are several ways to print out a value or string in MATLAB. We have already seen one: compute a value or just type a variable name at the prompt without a semicolon at the end and MATLAB will print out the value,

```
>> x = 1/3;
>> x
x =
    3.333333333333333e-01
```

(By the way, `format compact` has been used to get MATLAB to leave out the blank lines it would normally produce around `x =`).

Often we want to print out something more descriptive or make a table of values with numbers lined up nicely. The `disp` command prints out a variable without printing the name of the variable first,

```
>> x = 1/3;
>> disp(x)
    3.333333333333333e-01
```

`disp` can also be used to print a string of characters. In MATLAB, a string of characters is specified by putting them in single quotes,

```
s = 'Hello world';
disp(s)
```

or simply

```
disp('Hello world')
```

prints out this string.

You can build a string out of pieces by putting several strings together in an array. Here's a useful application of this:

```
>> s = ['The value of x is now ', num2str(x)];
>> disp(s)
    The value of x is now 0.33333
```

Note that `num2str` converts the number `x` into a string that can be catenated together with the other string to form the string `s`.

A more general way to make a string that includes one or more numbers when we want to control the spacing, number of digits printed out, etc., is to use the `sprintf` command, which writes formatted data to a string. This follows the C language formatting conventions if you are familiar with C. Type `help sprintf` for more details. Here is an example:

```
>> x = 1/3; y = 2/3;
>> s = sprintf(' We now have x = %20.15e and y = %10.5f', x,y);
>> disp(s)
  We now have x = 3.333333333333333e-01 and y =     0.66667
```

Here's an m-file script that approximates the value of the square root of a number given an initial guess. Note the use of the `input` command to read in data from the prompt:

```
% newtonsqrt.m
a = input('What number do you want to find the square root of? ');
x = input('What is your starting guess? ');
disp(' ');
disp('iteration approximation');
maxiter = 10;
% maximum number of iterations to allow
for i=1:maxiter
x = 0.5*(x + a/x);
disp(sprintf('%5i %20.15e', i,x));
if abs(x^2 - a) < 1e-12
break % breaks out of the for loop if x is good enough
end
```

```
end
if i==maxiter
disp('*** Warning: may not have converged -- tolerance not satisfied');
end
disp('For comparison,');
disp(sprintf(' sqrt(%g) = %20.15e', a, sqrt(a)));
```

Here is some sample output:

```
>> newtonsqrt
What number do you want to find the square root of? 2
What is your starting guess? 5

iteration approximation
  1  2.700000000000000e+00
  2  1.720370370370370e+00
  3  1.441455368177650e+00
  4  1.414470981367771e+00
  5  1.414213585796884e+00
  6  1.414213562373095e+00
For comparison,
  sqrt(2) = 1.414213562373095e+00
```

Note that this m-file is much fancier than the simple loop we used before. The core computation `x = 0.5*(x + a/x);` is the same as before, but we have added several features along with the tabular output:

- The program tests to see if x^2 is close to a each iteration, and if it is close enough we break out of the loop (this is a "convergence test").
- A maximum number of iterations (10 here) is specified so that we don't get into an infinite loop by accident.
- If the convergence test is never satisfied then a warning message is printed out so the user knows that the final `x` may not be sufficiently accurate.

Paying attention to these sorts of details is important and similar issues will come up with most of the numerical methods we study.

There is also a `fprintf` command that prints directly to a file. This is useful if you expect a lot of output and you want it to be output to a file instead of zipping past you on the screen. For example:

```
% fprintf_example.m
fid = fopen('outputfile.txt','w'); % open a text file for the output
for k=1:1000
  frpintf(fid, 'k = %4i\n', k); % note: \n means "new line"
end
fclose(fid);
```

produces a file named `outputfile.txt` containing 1000 lines

```
k = 1
k = 2
k = 3
k = 4
etc.
```

There are many other useful commands in MATLAB. Here are just a few examples:

- `A = zeros(m,n)` makes an $m \times n$ array of all zeros.
- `A = ones(m,n)` makes an $m \times n$ array of all ones.
- `size(A)` returns the dimensions of `A`, a row vector `[m n]`.
- `B = zeros(size(A))` makes an array of all zeros the same size as `A`.
- `A = rand(m,n)` makes an $m \times n$ array of random values, uniformly distributed between 0 and 1.
- `A = randn(m,n)` makes an $m \times n$ array of random values, normally distributed (with mean 0 and variance 1).
- `A = eye(n)` makes an $n \times n$ identity matrix.
- `d = diag(A)` when A is an $n \times n$ matrix, creates a vector containing oly the diagonal elements of A
- `A = diag(d)` when d is a vector of length n, creates an $n \times n$ diagonal matrix with $d(1), \ldots, d(n)$ on the diagonal.
- `sin, cos, tan, csc, sec, cot`: Trigonometric functions
- `asin, acos, atan, acsc, asec, acot`: Inverse trigonometric functions
- `ceil, floor, round, fix`: rounding real numbers to integers
- `max, min, mean, median, sum, prod`: act on vectors and return a scalar.

C Using Functions in Matlab

There are two types of m-files in MATLAB (files that end with the `.m` extension). One type is a **script**, which is just a list of commands that will be executed when you type the name of the file (without the `.m`) into MATLAB. The other type is a **function**, a file called `myf.m`, for example, which starts with a line like

```
function myf(x,y)
```

or

```
function f = myf(x,y)
```

or

```
function [f,g] = myf(x,y)
```

Each of these would take two inputs `x` and `y` (which might be single numbers, vectors, or matrices) and would presumably do some operations on the inputs. The first function would not return any values (though it might print something or produce a plot), the second form would return a single value (and `f` would have to be set somewhere in the body of the function). The third form would return two values `f` and `g` (both of which should be set somewhere in the body).

Many simple mathematical functions can be represented by a single formula in terms of elementary operations (addition, multiplication, etc.) or standard functions such as $\sin(x)$, $\cos(x)$, $\exp(x)$, $\sqrt{x}$, etc. An example would be

$$f_1(x) = 3x^4 + x\sin(2\pi x)/\sqrt{1+x^2}. \tag{C.1}$$

There are several possible ways to represent a function in MATLAB that allow us to evaluate the function at desired points. One approach is to create an m-file that implements the rule defining the function. For example, the function (C.1) can be implemented by creating a file `f1.m` containing the two lines

```
function y = f1(x)
y = 3*x.^4 + x.*sin(2*pi*x) ./ sqrt(1 + x.^2);
```

Then in MATLAB we can evaluate this in the obvious way:

```
>> z = f1(1)
z =
3
>> alpha = f1(-2)
alpha =
48
>> f1(.5)
ans =
0.1875
```

Note that we have implemented the formula for $f_1(x)$ in a vectorized manner using the operators `.*, .^ , ./` rather than simply `*, ^ , /`. If we only plan to evaluate `f1` at a single point, as in the above examples, then it doesn't matter which we use. However, by using the vector-component operators it is possible to compute $f_1(x)$ simultaneously for a whole vector of x values with a single call to `f1`. For example, we can compute f_1 at the three values used above by

```
>> f1values = f1([1 -2 .5])
f1values =
3.0000 48.0000 0.1875
```

Another example:

```
>> f1([1 2 3; 4 5 6])
ans =
3 48 243
768 1875 3888
```

This evaluates f_1 at a 2×3 array of points. It's a good idea to get in the habit of always using the vector-component operators in function definitions as it is often convenient to be able to evaluate a function at a whole set of points.

For example, suppose we want to plot the function $f_1(x)$ for $-1 \leq x \leq 1$ We can do this by evaluating f_1 at a large number of points in $[-1, 1]$ and then using the `plot` command in MATLAB:

```
>> x = linspace(-1, 1, 100);
>> y = f1(x);
>> plot(x,y)
```

After executing these commands, `x` and `y` are each vectors of length 100. The command `y = f1(x)` would result in a MATLAB error if we had used non-vectorized operators in the definition, since x^4, for example, is not defined unless `x` is a square matrix (and we don't want the matrix power). Another way to evaluate a function at a whole set of points is to use a loop,

```
>> x = linspace(-1, 1, 100);
>> y = zeros(size(x));
>> for i=1:length(x), y(i) = f1(x(i)); end
```

In this case `x(i)` is a scalar value and so `f1` is only called with scalar arguments. However, this is more cumbersome than simply setting `y = f1(x)` and is also typically much slower since 100 function calls are made instead of 1. For this small number the difference would be imperceivable, but for practical problems we will often need to evaluate functions thousands or millions of times and using vectorized form whenever possible can greatly reduce the computational time.

Note that in this last example we set `y = zeros(size(x))` before starting the loop that correctly evaluates `y(i)`. This preallocates storage for `y` of the correct size before looping through and resetting each component of the array.

This is not strictly necessary for this example: MATLAB will automatically increase the size of the array `y` each time through the loop if necessary as we set each value. However, it is often more efficient to preallocate the storage since resizing `y` repeatedly is time consuming (and programs that do this will run more slowly). Another reason for preallocating the storage is that it insures that `y` will be the size we expect. If `y` has never been used before in this MATLAB session then we'll get what we expect without preallocating, but if `y` has already been used for some other purpose then we it may be the wrong size. For example, suppose `y` was previously used to store a vector of 500 values. Then the commands

```
>> x = linspace(-1, 1, 100);
>> for i=1:length(x), y(i) = f1(x(i)); end
```

would reset the first 100 elements of `y` but would not change the remaining 400 elements. The command `plot(x,y)` would then produce an error message since `x` and `y` are not the same size.

C.1 In-line function definitions

For functions that are implemented by a single MATLAB statement, such as `f1` above, there is another way to specify the function rather than creating a 2-line m-file as done above. It is often more convenient to define it directly at the MATLAB command line as an in-line function, as follows:

```
>> f1a = @(x) 3*x.^4 + x.*sin(2*pi*x) ./ sqrt(1 + x.^2);
```

The syntax indicates that `f1a` is a function of a single variable that is denoted by `x` in the formula that follows. Once `f1a` has been defined in this way, it can be evaluated in the same manner as m-file functions:

```
>> f1a(.5)
ans =
0.1875
```

The definition of a function in this manner can use other variables,

```
>> a = 25;
>> g = @(x) a + x;
>> g(4)
ans =
29
>> a = 100;
>> g(4)
ans =
29
```

When `g` is defined `a` has the value 25 and this value is then hard-wired into the function definition. The syntax `g = @(x) ...` indicates that the function is

being defined as a function of a single variable denoted by `x` in the following string, and so `a` is assumed to be some value already set before executing this function definition (an error message will result if `a` has not already been set). Note that later changing `a` without redefining `g` will not change the function, it as if we had defined

```
>> g = @(x) 25 + x;
```

Functions of more than one variable can be defined by specifying more than one argument,

```
>> z = 20;
>> g = @(x,y) x.*y + z*sin(y);
>> g(2,pi/2)
ans =
2.314159265358979e+01
```

This implements the function $g(x, y) = xy + 20\sin(y)$. Again note that the notation defining `g` indicates that it is a function of the two variables `x` and `y` in the formula following, and so `z` is assumed to be a previously defined value.

C.2 Passing function names into other functions

Sometimes we will want to define or use functions that act on other functions. For example, the built-in MATLAB function `quad` computes numerical approximations to definite integrals, $\int_a^b f(x)\,dx$. (This sort of approximation, which we'll study later, is sometimes called *quadrature*, which explains the name of the MATLAB function.) To apply `quad`, we have to give it the function f and also the endpoints of the interval a and b. Suppose for example we want to approximate $\int_0^1 f_1(x)\,dx$, where $f_1(x)$ is the function specified in (C.1). Then we pass in to `quad` an implementation of the function f_1 and the endpoints. We have seen two different ways to implement f_1 above, either as an m-file or as an in-line function. Either can be used, but the syntax is slightly different:

```
>> quad('f1',0,1)
ans =
4.864166380749058e-01
>> quad(f1a,0,1)
ans =
4.864166380749058e-01
```

If we use an m-file to define f_1, then we pass the name of the m-file to `quad` as a string, and hence it is surrounded by quotes. If we use the in-line version then `f1a` is a variable in MATLAB (of type "function") and so we just pass in this variable (without quotes).

D Plotting Functions in Matlab

Manually, when we want to plot a curve, like $y = \sin(x)$ over $[0, \pi]$, we start by taking arbitrary values for x in $[0, \pi]$ and computing the corresponding value for y. We plot the resulting points (x, y) and connect the points by a straight line, as in "connect the dots" drawing.

- In "connect the dots" drawing, the dots are ordered by their number. Here the chosen order is increasing (or decreasing) values of x.
- We need to specify enough points in $[0, \pi]$ so that the sequence of lines connecting the points will give a graph that is visually indistinguishable from the plot of $f(x)$.

In MATLAB, plotting a curve follows the same procedure. We define a vector `x` containing many points in the interval of interest, evaluate `y = f(x)` as a vector of the corresponding function values, and then execute `plot(x,y)`. The `plot` command produces a graph by connecting the points specified by straight lines, as in a "connect the dots" drawing.

If $f(x)$ is a smooth function and we specify a dense enough set of points, then this sequence of straight lines will give a graph that is visually indistinguishable from a plot of the function $f(x)$. But it is important to understand what MATLAB is doing because it is easy to produce graphs that do a very bad job of illustrating the true function if we are not careful, particularly if the function f is not very smooth or has discontinuities. We will explore this with a number of examples.

Consider $y = \sin(x)$ over $[0, \pi]$.

1. Create sampling values in $[0, \pi]$ and store them in a column vector **x**.

```
>> x = []; m = 25;
>> for i = 1:(m+1),
x = [x; (i-1)*pi/m];
end
>> x

x =

         0
    0.1257
    0.2513
    0.3770
     .....
```

2. Plot and connect the dots

```
>> plot(x,sin(x))
```

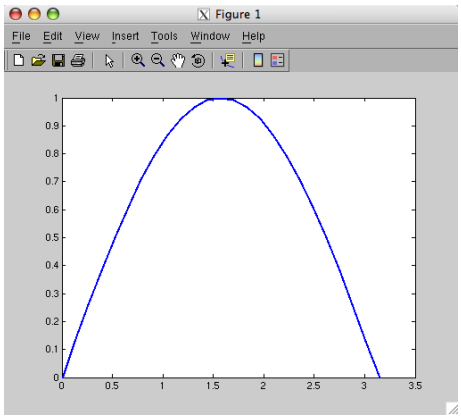


Figure D.1: Figure for `plot(x,sin(x))`.

Consider now $y = x^2$ over $[0, \pi]$.

```
>> x = [0; 0.5; 1.0; 1.5; 2.0; 2.5; 3.0; pi];
>> plot(x, x*x)
```

The command result in an error because the vector **x** can not be multiplied by a vector in linear algebra sense. Instead, we want a component-wise multiplication, resulting in a vector with 8 components containing values of x^2. To get a component-wise version, we type

```
>> x = [0; 0.5; 1.0; 1.5; 2.0; 2.5; 3.0; pi];
>> plot(x, x.*x)
```

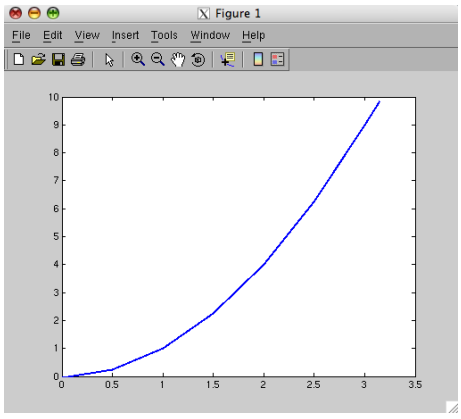


Figure D.2: Figure for `plot(x, x.*x)`.

The same modification applies for the division operator / or the power operator ^.

Order of points. The order of points in the sampling vector $\mathbf{x}$ defines the order when connecting the dots. Figure D.3 presents two plots with two different orderings in the vector $\mathbf{x}$:

1. $\mathbf{x} = \begin{bmatrix} 0 \\ 3 \\ 1 \\ 2 \end{bmatrix}$

2. $\mathbf{x} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$

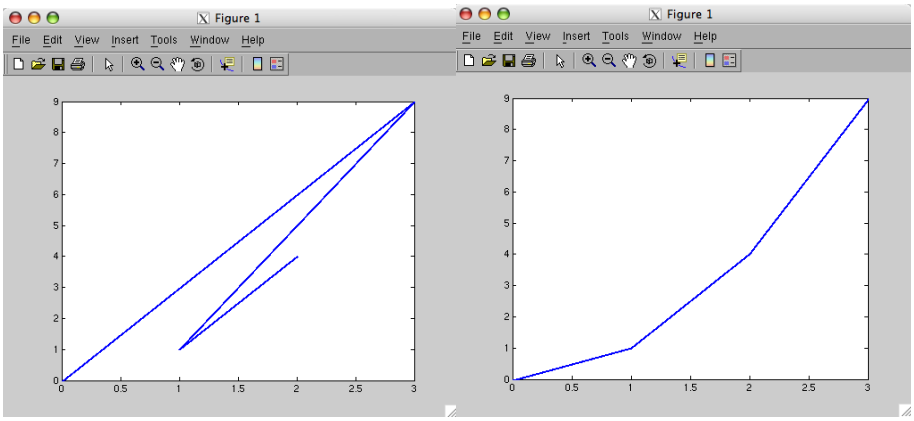


Figure D.3: Plots with different order of points in $\mathbf{x}$

Parametrized curve. Consider now the curve $x^2 + 4y^2 = 1$. It can be parametrized as follows

$$x = \cos(t), \quad y = \frac{\sin(t)}{2}, \quad \forall t \in [0, 2\pi].$$

The procedure to plot this curve with MATLAB goes

1. Create a vector sampling $[0, 2\pi]$.

```
>> t = linspace(0, 2*pi, 100)';
```

2. Plot the curve

```
>> plot(cos(t), sin(t)/2)
```

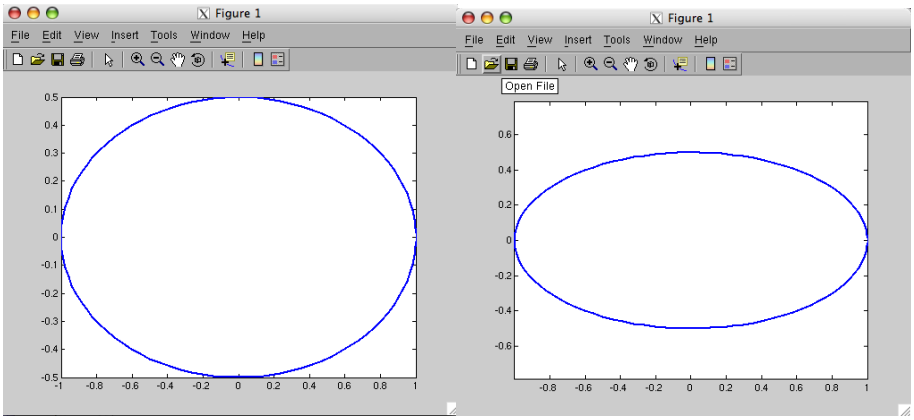


Figure D.4: Parametrized curve with default view (left) and with equal axes (right).

Suppose we want to plot

$$f(x) = 2\sin(x) + \cos(5x) \tag{D.1}$$

for $0 \leq x \leq 10$ and have defined a suitable m-file or inline function `f` that evaluates this function. Figure D.5 shows the result of the following commands:

```
>> x = linspace(0, 10, 1000);
>> plot(x,f(x))
```

This gives a reasonable graph of this function. Note that it consists of a rapidly varying oscillation (the $\cos(5x)$ term) superimposed on a slower oscillation (the $2\sin(x)$ term). With 1000 points the curve plotted looks smooth and captures the function well. If we increased the number of points, to 5000 say, the plot would look essentially the same. However, if we reduce the number of points and used only 30, then we would obtain the graph of Figure D.5(b), plotted using

```
>> x = linspace(0, 10, 30);
>> plot(x,f(x))
```

This still captures the essential behavior of the function, but it does not look so smooth — it's now clear that linear segments are plotted between specified points. If we reduce the number of points used further, to only 10, we obtain the plot of Figure D.5(c). This looks smoother but doesn't look much like the proper graph. The reason why is seen in Figure D.5(d), where the particular points chosen are shown as dots along with the plot of the proper function.

The collection of straight lines plotted by the `plot` command can be viewed as the the correct graph of a *piecewise linear function* that approximates the function $f(x)$ that we really intended to plot. The idea of approximating a

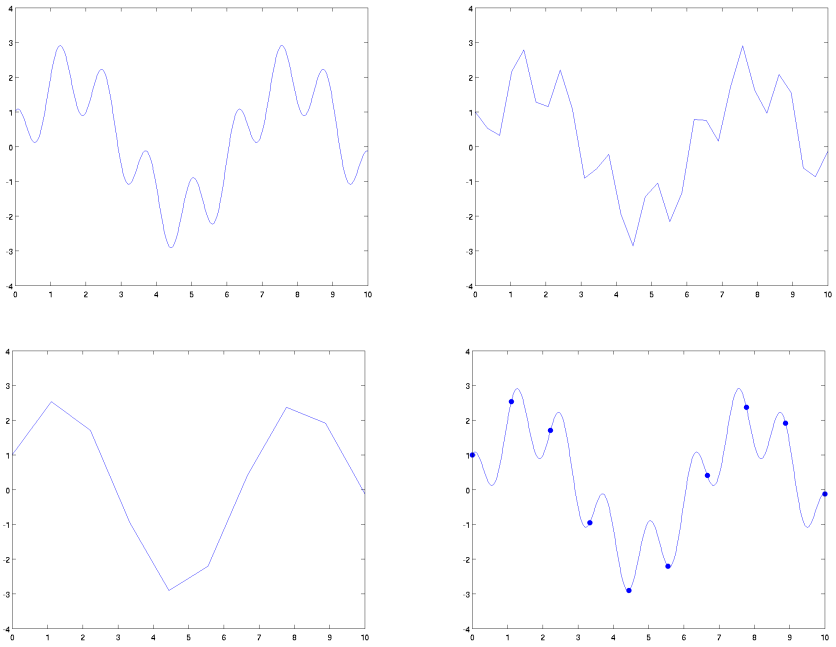


Figure D.5: (a) Plot of function (D.1) using 1000 points. (b) Plot of function (D.1) using 30 points. (c) Plot of function (D.1) using 10 points. (d) Illustration of where these 10 points lie on the curve $y = f(x)$.

complicated function by a much simpler function is one of the main themes of this course and we will see many applications of this idea. MATLAB uses this approximation to plot a set of points since graphics primitives exist to connect any two points by a straight line. We will use this idea in other contexts because linear functions are much easier to work with than arbitrary functions in many ways. For example, we can easily integrate a linear function and so if we approximate a function by a piecewise linear approximation we can approximate the integral.

As another example of potential pitfalls when plotting, consider the function

$$f(x) = x(3 - x) + \exp(-100|x - 1|). \tag{D.2}$$

Suppose we plot this function over the interval $[0, 3]$ using the commands

```
>> f = @(x) x.*(3-x) + exp(-100*abs(x-1));
>> x = linspace(0,3,30);
>> plot(x,f(x))
```

Then we obtain the plot shown in Figure D.6(a). There's a little bump in the curve near $x = 1$ but it's barely noticeable and if we don't think about it we might conclude that we have a pretty good plot of $f(x)$ using only 30 points. However, note that $f(1) = 2 + \exp(0) = 3$. This doesn't agree with what is shown in the plot at all! The function has an exponentially decaying spike centered at $x = 1$. To capture this we must plot using many more points. For example, Figure D.6(b) shows a plot with 1000 points.

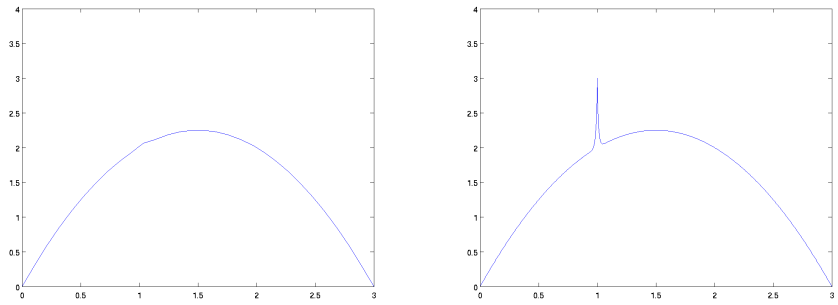


Figure D.6: (a) Plot of function (D.2) using 30 points. (b) Plot of function (D.2) using 1000 points.

For the function $f(x)$ in (D.2) it is no big deal to evaluate the function at 1000 points to produce this plot — it takes a small fraction of a second in MATLAB. However, for other practical problems it may be quite expensive to compute each function evaluation (for example when each "function evaluation" requires going off and solving some other problem). If so then we might want to give more thought to where we really need to evaluate f many times and where

a more coarse representation of f suffices. In the example we've just considered, it is only near $x = 1$ that we need lots of values of f in order to capture the behavior. Away from this point we only need a few values to get a good looking plot of the quadratic behavior. We can produce a plot that looks just as good as Figure D.6(b) but using far fewer evaluations of the function f by using an unequally spaced set of points, *e.g.*,

```
>> x = [linspace(0,0.9,10)  linspace(0.9,1.1,200)  linspace(1.1,3,20)];
>> plot(x,f(x))
```

The vector `x` used here has 10 points between 0 and 0.9, followed by 200 points between 0.9 and 1.1, and finally 20 points between 1.1 and 3, for a total of 230 points rather than 1000.

This example serves as a warning that it is good to know something about the function you are trying to plot in order to obtain a reasonable result. This warning applies to many of the numerical methods we will consider as well, even if you are simply using built-in MATLAB commands. Do not always blindly trust what the computer tells you. For example, suppose we want to compute $\int_0^3 f(x)\,dx$ for the function of (D.2). We might again try the MATLAB function `quad` and we would obtain

```
>> quad(f,0,3)
ans =
     4.500000072864403e+00
```

We might conclude that 4.5000 is an accurate approximation to the integral.

However, for this particular function we can integrate exactly and find that

$$\begin{aligned}\int_0^3 x(3-x) + \exp(-100|x-1|)\,dx &= \left[\frac{3}{2}x^2 - \frac{1}{3}x^3\right]_0^3 + \int_0^3 \exp(-100|x-1|)\,dx \\ &= 4.5 + \int_0^1 \exp(-100x)\,dx + \int_0^2 \exp(-100x)\,dx \\ &= 4.5 + 0.01(1-\exp(-100)) + 0.01(1-\exp(-200)) \\ &\approx 4.520000000000000000\ldots\end{aligned} \tag{D.3}$$

The value 4.5000... returned by `quad` has an error of about 0.02 which is quite significant. It's easy to see where this error comes from — `quad` is fooled in much the same way we might be by Figure D.6. It has entirely missed the sharp peak near $x = 1$ and has produced a good approximation to $\int_0^3 x(3-x)\,dx$.

Note that we can get a much better approximation using `quad` if we split the integral up into an integral from 0 to 1 and one from 1 to 3. But splitting the interval at the point where the spike lies, we make sure that `quad` notices something is going on at this point:

```
>> quad(f,0,1) + quad(f,1,3)
ans =
     4.520000793872945e+00
```

D.1 Some other useful plotting commands

The following commands are often useful. Use the MATLAB help facilities for more information on plotting.

- `axis([a b c d])` Set the region shown to $a \leq x \leq b$ (horizontal axis) and $c \leq y \leq d$ (vertical axis).
- `hold on` Hold the current plot so that the next plot command will appear together with the current plot. Useful for plotting several curves together or points on top of curves. Turn this off with `hold off` or use the `clf` (clear frame) command to return to a black frame.
- `figure(2)` Open a second plot window, or bring this one to the top.