Linear Algebra, First Part
Lecture Notes

Bernd Gärtner

September 7, 2024

Contents

Chapter 0

Introduction

0.1 About these notes

These are the lecture notes for the first part of the course

Lineare Algebra (401-0131-00L)

held at the Department of Computer Science at ETH Zürich in HS24.

These notes can be considered as a full version of what I plan to write on the tablet during the lectures. Lecture plans will be made available before each lecture. The tablet notes (in German) reflect the reality and will be made available after each lecture. Together with the explanations (and answers to questions) given in the lectures, they will contain the "essentials", but in order to fully understand them, it can be helpful to look up more details and additional explanations in the lecture notes.

In content, the lecture notes are based on the book

> *Introduction to Linear Algebra* (Sixth Edition) by Gilbert Strang, Wellesley - Cambridge Press, 2023 [Str23].

The main difference is that the approach taken here is more rigorous than Strang's. Strang introduces the material on an intuitive level, guided by many examples; this provides a great informal introduction to Linear Algebra. What we add here (hopefully without losing the intuition) are formal definitions of concepts, as well as mathematical statements with proofs for the key results.

Strang's book is *not* part of the course's official material, and there is no need for students to buy the book (it doesn't have an official electronic version). With the lectures, lecture plans, tablet notes, lecture notes, exercises, and exercises classes, the course is self-contained. Strang's book and others mentioned on the course web page serve as optional literature.

Zürich, September 3, 2024 Bernd Gärtner

0.2 About computer science (and mathematics)

As a first semester student of computer science, you may wonder why one of the first things you see is mathematics, and in particular linear algebra. In order to answer this, we first need to answer a different question: *What is computer science?*

Computer scientists are frequently approached for help with installing computers, getting the internet to work, and similar technical tasks. To many people, a computer scientist is simply an information technology expert. But this point of view is fundamentally wrong. The computer scientist Mike Fellows has explained this very well already in 1991, using a simple analogy:

> Computer science is not about machines in the same way that astronomy is not about telescopes. There is an essential unity of mathematics and computer science [Fel93].

The first sentence of this quote (often misattributed to Edward Dijkstra) is well-known and gets the main point across: computer science is *not* mainly about computers. Everyone agrees that computers, just like telescopes, are great tools, but they merely help us in achieving some goals, they are not the goals themselves. It is true that astronomers are involved in building telescopes, and computer scientists are involved in building computers. Generally, if you need a tool that you cannot buy off the shelf, you will team up with people that can build it for you, and this needs expertise from both sides. But in the end, you want to use the tool for *something*, so you are not mainly interested in the tool itself.

The second part of the quote is less known, but not less important. It indicates that computer science and mathematics are very strongly connected. To understand this, we need to say what computer science *actually* is. If you ask the internet for a definition of computer science, you get many wrong answers that start with "the study of computers…", even from serious sources. The Wikipedia article about computer science starts differently and does not mention computers in the first sentence:[1]

> Computer science is the study of computation, information, and automation.

This is correct but a bit too short; the German version of the page,[2] has a more detailed and clearer definition, taken from the "Duden Informatik A-Z" [CSB06, Eintrag Informatik, S. 305].

> Informatik ist die Wissenschaft von der systematischen Darstellung, Speicherung, Verarbeitung und Übertragung von Informationen, besonders der automatischen Verarbeitung mit Hilfe von Digitalrechnern.

[1] `https://en.wikipedia.org/wiki/Computer_science`, accessed on August 2, 2024

[2] `https://de.wikipedia.org/wiki/Informatik`, accessed on August 2, 2024

In English, this reads as follows:

> Computer science is the science of the systematic representation, storage, processing and transfer of information, in particular of the automatic processing using computers.

This means, computer science is about dealing with *information*. The German term *Informatik* transports this message much better than the English term *Computer science*.

As an example, let's consider addition as you (and children throughout many centuries) have learned it in primary school, for example

$$\begin{array}{cr} & 123 \\ + & 486 \\ \hline = & 609 \end{array}$$

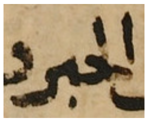
This is systematic processing of information (in this case numbers), represented in decimal place-value system. Hence, this is not only mathematics, but according to the above definition, it is also computer science! In modern terms, we would call schoolbook addition an algorithm. Computers can do such additions automatically and very fast (this is what the "in particular" part of the definition is about), but the algorithm itself was invented much earlier than the computers.

While an algorithm is mostly associated with computer science, the theoretical foundations of many algorithms and other computer science inventions are inherently mathematical. Schoolbook addition is a prime example. Here, the most important theoretical foundation is the place-value system. This is one of the great historical developments in mathematics, driven by the need to efficiently compute with numbers.

Today, there are new needs, for example, efficiently training huge machine learning models, or securing computer systems against cyberattacks. This needs established as well as new mathematics. Mathematical research is often motivated by applications in computer science, and mathematicians work with computer science tools on a daily basis. For these reasons, every computer science student needs mathematical foundations, and every mathematics student needs computer science foundations. An essential unity, indeed.

0.3 About linear algebra

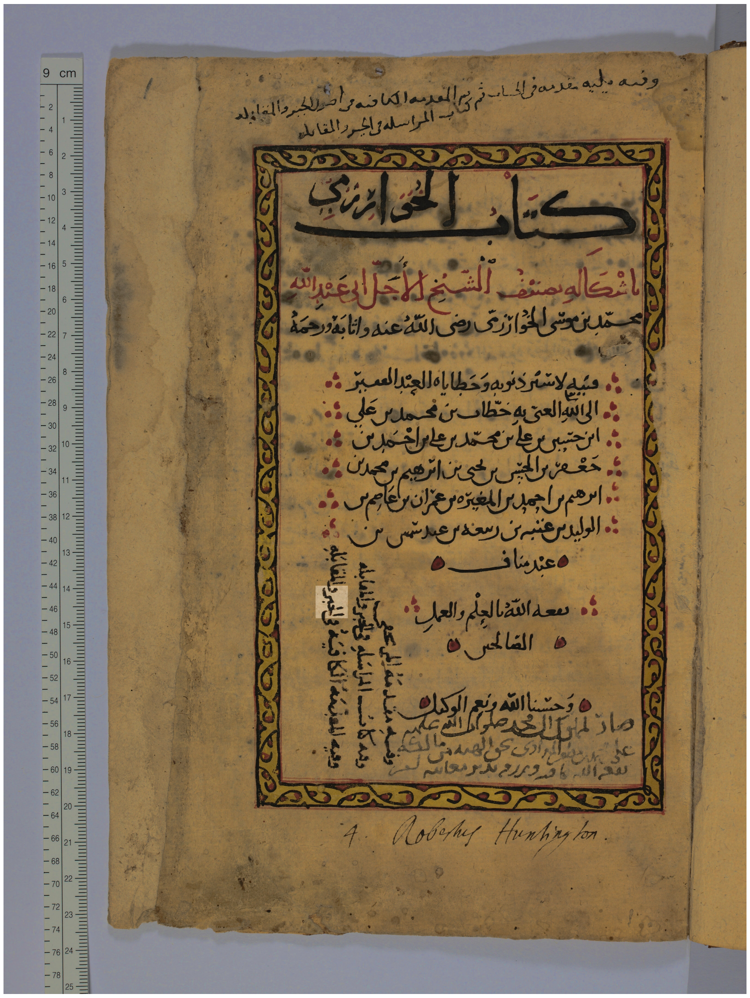
The origins of (linear) algebra can be traced back to the 9th century when the Persian polymath Al-Khwarizmi published *The Compendious Book on Calculation by Completion and Balancing*. The word *algebra* also goes back to this book, see the highlighted part on the title page shown in Figure 1. In modern Arabic letters and transcribed to Latin letters (right to left), this reads as follows:



= الجبر =

ـر	ـبـ	ـجـ	لـ	ا
r	b	j	l	A

Al Khwarizmi's book teaches how to systematically solve linear and quadratic equations in one variable. While this is basic highschool material today, the theory underlying it had to be developed at some point, and it was Al-Khwarizmi who did this; for example, the word *balancing* in the title of his work refers to the technique of moving a term to the other side of an equation, something you need to do when you want to solve for a variable.



وفيه يليه مقدمة في الحساب ثم يتم المقدمة الكافية في اصل الجبر والمقابلة
كتاب المراسلة في الجبر والمقابلة

كتاب الخوارزمي
باشكاله مصنف الشيخ الأجل ابي عبد الله
محمد بن موسى الخوارزمي رضي الله عنه واثابه ورحمة
وقفه لاستر ذنوبه وخطاياه العبد الفقير
الى الله الغني به خطاب بن محمد بن علي
ابن حسين بن علي بن محمد بن علي بن احمد بن
جعفر بن الحسين بن يحيى بن ابرهيم بن محمد بن
ابرهيم بن احمد بن المغيرة بن عمران بن عاصم بن
الوليد بن عتبة بن ربيعة بن عبد شمس بن
عبد مناف
نفعه الله بالعلم والعمل
الصالحين
وحسبنا الله ونعم الوكيل

وفيه المقدمة الكافية في الجبر والمقابلة
وفيه كتاب المراسلة في الجبر والمقابلة

4. Robertus Huntington.

Figure 1: Title page of Al-Khwarizmi's book; the highlighted text in the lines written from bottom to top is the Arabic word *Al-jabr*. This is a higher-resolution image of the one found on Wikipedia (`https://en.wikipedia.org/wiki/Al-Jabr`), provided to the author by Digital Bodleian, `https://digital.bodleian.ox.ac.uk`, CC-BY-NC 4.0. A translation of the title page can be found here

The field of linear algebra has developed from two historical roots: *analytic geometry* and *linear equations*. Analytic geometry deals with the description of and calculation with geometric objects through coordinates and formulas. We could also call it "rigorous"

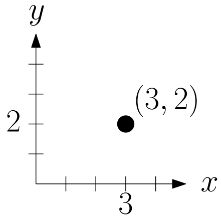


Figure 2: A point in the two-dimensional plane, expressed in Cartesian coordinates

geometry. Everybody knows what a circle is, but if you want to describe a particular circle, you need to say what the center and the radius are. The center is a point that you typically describe with *Cartesian coordinates*, as in Figure 2.

Having such rigorous descriptions of geometric objects, you can start to answer questions about them *analytically* by using mathematics; for example, do two given lines in three-dimensional space intersect or not?

In answering this and many other questions, systems of linear equations in more than one variable come up, and this is the second root of linear algebra. Today, such systems are considered as easy to solve in many cases, but this is the result of developments that happened over centuries. And they still happen now: solving systems of linear equations is an active research topic, in particular since the traditional algorithms cannot handle the very large systems that we have in many applications today. Even small systems are not necessarily easy for a human and form a basis of many puzzles. Consider this one:

> Dominik is twice as old as Susanne and three years older than Claudia. Together, the children are 17 years old. How old are the three children?

Even without having heard about linear equations, you will be able to figure this out, employing some guesswork and mental arithmetic. But this quickly reaches its limits when more children are involved in the puzzle. There is a reason why such puzzles that are made for entertainment rarely have more than three variables (in this case, the ages of the children). As a system of three linear equations, the puzzle reads as follows:

$$\begin{array}{rcl} D & = & 2S \\ D & = & C+3 \\ D+S+C & = & 17 \end{array}$$

Here, D, S, and C are variables for the unknown ages of the children, and the three equations encode the three pieces of information that the puzzle provides. The equations are *linear* because every variable occurs only in the first power. In contrast, $x^2-2x+3=0$ is a quadratic equation, because the variable x occurs in the second power as well. There are also cubic, quartic, quintic, etc. equations. Solving systems of those falls into the domain of (non-linear) algebra. Linear algebra deals with m linear equations in n variables, where both numbers m and n can be very large in practice.

In the above linear puzzle equations, you can convince yourself that there are unique numbers that you can plug in for D, S, and C such that all three equations are satisfied. Conveniently, these numbers are natural numbers; you don't want a puzzle where a child is 2.7 years or -3 years old. But in principle, the solutions to a system of linear equations can be arbitrary numbers, and it is up to the application to decide whether these make sense or not.

Interestingly, Al-Khwarizmi only provided formulas for positive solutions of equations in his book, as he thought that only those make sense. Indeed, in a world where numbers count physical quantities, the idea of a negative number seems downright crazy. However, in Al-Khwarizmi's computations (including "balancing"), negative numbers implicitly occur in order to arrive at the positive solutions.

Starting from the roots of analytic geometry and systems of linear equations, linear algebra has grown many important branches, some of which appear in Figure 3 and also later in these notes.

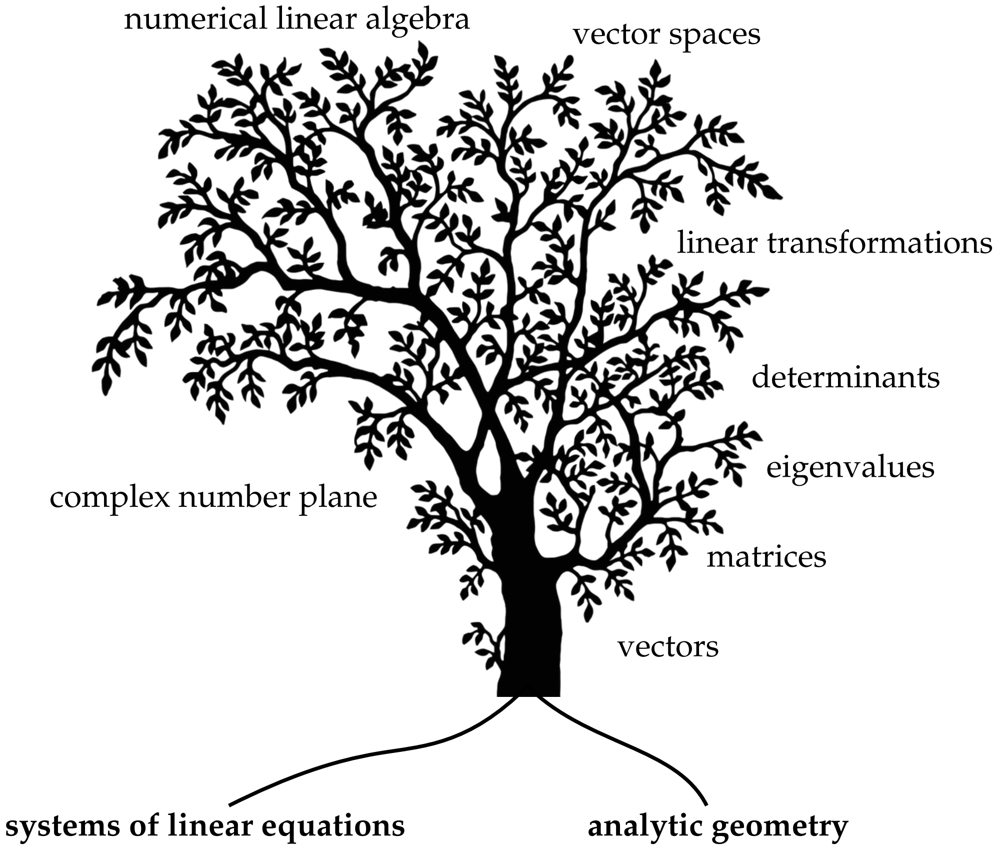


Figure 3: The tree of linear algebra: Roots and some important branches

Chapter 1

Vectors

1.1 Vectors and linear combinations

Vectors in $\mathbb{R}^m$ and their linear combinations are fundamental in linear algebra and at the same time easy to understand. In fact, you may know much of this material from highschool. This leaves room to also learn about some important elements of mathematical thinking and writing. Most notably, you will see a first proof.

Vectors as you know them from highschool "live" in 2-dimensional or 3-dimensional space. More abstractly, they live in some m-dimensional space, where $m \in \mathbb{N}$, the set $\{0, 1, 2, \ldots\}$ of *natural numbers*.[1] Starting from $m = 4$, these spaces are hard to visualize, but linear algebra can handle them as easily as $\mathbb{R}^2$ and $\mathbb{R}^3$.

Mathematically, these spaces are sets that are called $\mathbb{R}^2, \mathbb{R}^3$, and $\mathbb{R}^m$, where $\mathbb{R}$ is the set of *real numbers*. $\mathbb{R}^2$, the xy-plane, contains (has as elements) all pairs (v_1, v_2) of real numbers, for example $(4, 1)$. $\mathbb{R}^3$, the xyz-space, contains all triples (v_1, v_2, v_3) of real numbers, for example $(-2, 2, 3)$. $\mathbb{R}^m$ contains all *tuples* or *sequences* $(v_1, v_2, \ldots, v_m)$ of m real numbers. Here, the numbers from 1 to m serve as *indices* indicating the position in the sequence: for example, v_5 denotes the 5-th number in the sequence.

When we think of the elements of $\mathbb{R}^2$ or $\mathbb{R}^3$ as vectors, we typically draw them as arrows in a *Cartesian coordinate system*, with the tail of the arrow at the origin, and the head at the respective coordinates. We use *column vector* notation, indicating that we now think of a pair (or triple, or tuple) as a vector; see Figure 1.1. Sometimes, we use *row vector* notation as in $\begin{bmatrix} 4 & 1 \end{bmatrix}$. In referring to a vector in text or in a formula, we use bold lower case Latin letters such as $\mathbf{v}$ and $\mathbf{w}$. You may have learned to write vectors as $\vec{v}, \vec{w}$, but $\mathbf{v}$ is as good (or bad) as $\vec{v}$. The important thing is to be consistent.

The *zero vector* corresponds to the origin and is written as $\mathbf{0}$, in every dimension. This is what mathematicians call an *abuse of notation*. The abuse here is that the meaning of $\mathbf{0}$

[1] For mathematicians, 1 is typically the first natural number; for computer scientists it's 0. None of the two choices is better or worse than the other one. In these "linear algebra for computer scientists" notes, we start with 0.

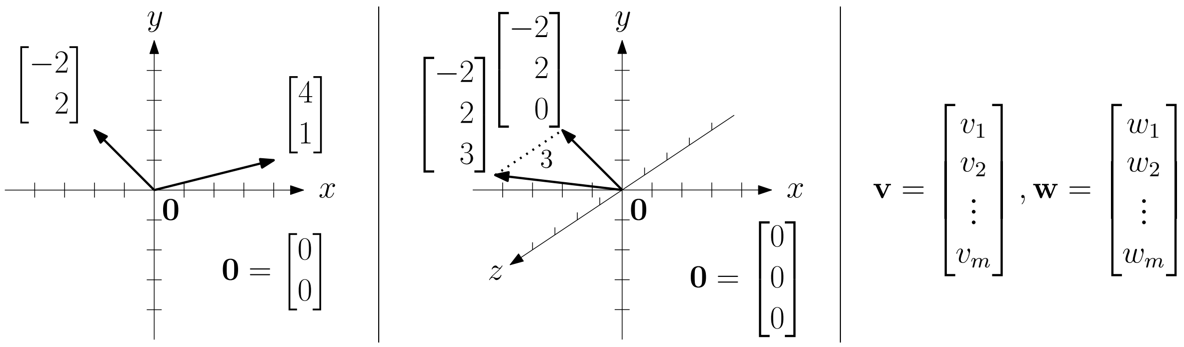

y
$\begin{bmatrix}-2\\2\end{bmatrix}$
$\begin{bmatrix}4\\1\end{bmatrix}$
x
$\mathbf{0}$
$\mathbf{0}=\begin{bmatrix}0\\0\end{bmatrix}$
$\begin{bmatrix}-2\\2\\0\end{bmatrix}$
$\begin{bmatrix}-2\\2\\3\end{bmatrix}$
3
z
$\mathbf{0}=\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\mathbf{v}=\begin{bmatrix}v_1\\v_2\\\vdots\\v_m\end{bmatrix},\mathbf{w}=\begin{bmatrix}w_1\\w_2\\\vdots\\w_m\end{bmatrix}$

Figure 1.1: $\mathbb{R}^2$, the xy-plane $\qquad$ $\mathbb{R}^3$, the xyz-space $\qquad$ Column vectors in $\mathbb{R}^m$

depends on the context and may, as in Figure 1.1, refer to the zero vector in $\mathbb{R}^2$ or the zero vector in $\mathbb{R}^3$. Such abuse of notation is common and also known from natural language. If you take "the bike", it is clear that you mean your bike, while your friends, saying the same thing, refer to their bikes.

The arrow drawing suggests an interpretation of a vector as a movement, for example "go 4 steps right and 1 step up!" Under this interpretation, the arrow can actually be placed anywhere and still visualizes the same vector. It can also be useful to visualize a vector as a point (at its coordinates); see Figure 1.2.

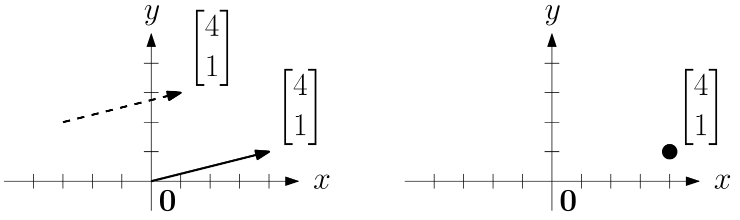

y
$\begin{bmatrix}4\\1\end{bmatrix}$
$\begin{bmatrix}4\\1\end{bmatrix}$
x
$\mathbf{0}$
y
$\begin{bmatrix}4\\1\end{bmatrix}$
x
$\mathbf{0}$

Figure 1.2: Vector: visualization as arrow (left), or point (right)

As an example why this is useful, let's try to visualize the set of vectors in $\mathbb{R}^2$ whose coordinates sum up to 5, highlighting a specific such vector. Then it should be clear that Figure 1.3 (right), with the gray line showing the vectors in question as points at their arrowheads, is more suitable than Figure 1.3 (left) that is just cluttered with arrows.

1.1.1 Vector addition

Adding two vectors combines their movements. Depending on which movement we do first, we can take two different routes to the same result; together, these routes form a *parallelogram*. Algebraically, *vector addition* works *coordinate-wise*, i.e. we add up corresponding coordinates of the two vectors; see Figure 1.4.

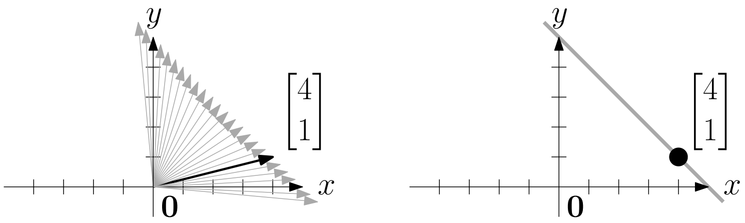
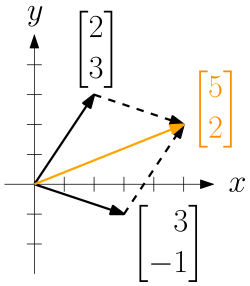

y
$\begin{bmatrix}4\\1\end{bmatrix}$
x
0
y
$\begin{bmatrix}4\\1\end{bmatrix}$
x
0

Figure 1.3: Vectors $\mathbf{v}$ in $\mathbb{R}^2$ with $v_1 + v_2 = 5$: visualization as arrows (left), or points at their arrowheads (right)


y
$\begin{bmatrix}2\\3\end{bmatrix}$
$\begin{bmatrix}5\\2\end{bmatrix}$
x
$\begin{bmatrix}3\\-1\end{bmatrix}$

$$\begin{bmatrix}2\\3\end{bmatrix} + \begin{bmatrix}3\\-1\end{bmatrix} = \begin{bmatrix}5\\2\end{bmatrix}$$

Figure 1.4: The parallelogram of vector addition

Definition 1.1 (Vector addition)**.** *Let*

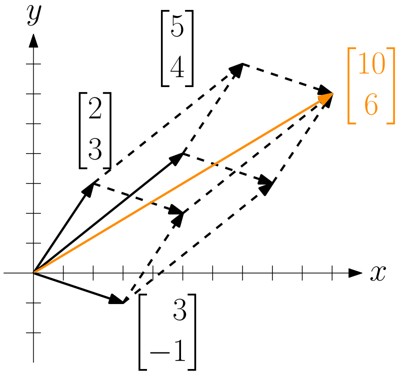
$$\mathbf{v} = \begin{bmatrix}v_1\\v_2\\\vdots\\v_m\end{bmatrix}, \mathbf{w} = \begin{bmatrix}w_1\\w_2\\\vdots\\w_m\end{bmatrix} \in \mathbb{R}^m. \textit{ The vector } \mathbf{v}+\mathbf{w} := \begin{bmatrix}v_1+w_1\\v_2+w_2\\\vdots\\v_m+w_m\end{bmatrix} \in \mathbb{R}^m \textit{ is the } \text{sum} \textit{ of } \mathbf{v} \textit{ and } \mathbf{w}.$$

While examples are very useful to understand a concept, a *definition* is the official reference. The goal of a definition is to introduce the concept in full generality. Definition 1.1 tells you how to add *any* two vectors of *any* dimension. The symbol $:=$ is used to define what's left of it by what's right of it.

In the same way, we can add more vectors. For example $\mathbf{u}+\mathbf{v}+\mathbf{w}$ is the sum of three vectors, and in this case, we get six possible routes to the result; see Figure 1.5.

Unfortunately, Definition 1.1 does not define $\mathbf{u}+\mathbf{v}+\mathbf{w}$. It only talks about adding *two* vectors at a time. So it defines $(\mathbf{u}+\mathbf{v})+\mathbf{w}$ or $\mathbf{u}+(\mathbf{v}+\mathbf{w})$. Since addition in $\mathbb{R}$ is *associative*, it doesn't matter where we put the brackets, so it is customary to omit them and write $\mathbf{u}+\mathbf{v}+\mathbf{w}$. Similarly, in talking about the two possible routes to arrive at $\mathbf{v}+\mathbf{w}$, we implicitly used that $\mathbf{v}+\mathbf{w} = \mathbf{w}+\mathbf{v}$ which holds since addition in $\mathbb{R}$ is *commutative*. The six possible routes for three vectors come from the fact that we can add them up in six different orders.

It is an element of mathematical thinking to clarify the meaning of $\mathbf{u}+\mathbf{v}+\mathbf{w}$ when this was not explicitly defined. Only after that, you *really* understand the meaning and can


y
$\begin{bmatrix}5\\4\end{bmatrix}$
$\begin{bmatrix}2\\3\end{bmatrix}$
$\begin{bmatrix}10\\6\end{bmatrix}$
x
$\begin{bmatrix}3\\-1\end{bmatrix}$

$$\begin{bmatrix}2\\3\end{bmatrix}+\begin{bmatrix}3\\-1\end{bmatrix}+\begin{bmatrix}5\\4\end{bmatrix}=\begin{bmatrix}10\\6\end{bmatrix}$$

Figure 1.5: Adding three vectors

safely write $\mathbf{u}+\mathbf{v}+\mathbf{w}$. As a Swiss luxury watch commercial puts it: to break the rules, you must first master them.

Figure 1.6 illustrates some of the routes one can go in adding up 9 vectors. We haven't drawn all possible routes, as this would lead to a very cluttered figure.

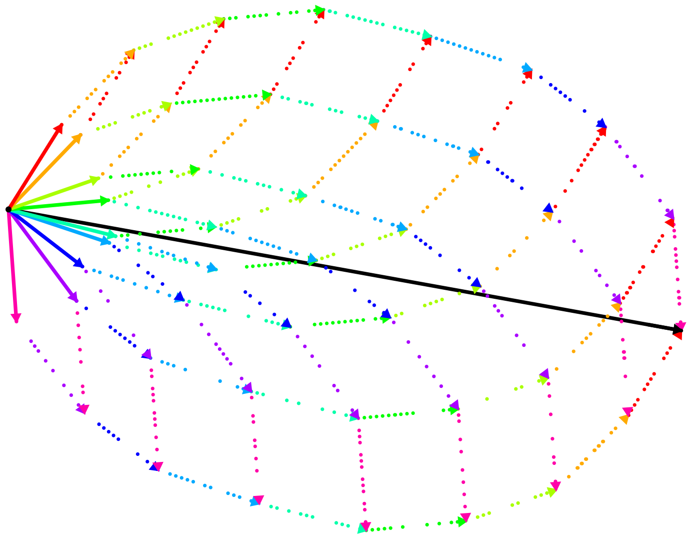


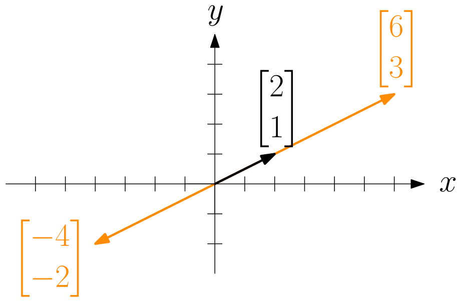
Figure 1.6: Adding 9 vectors (solid colored arrows); the result is the black arrow, and the dashed arrows outline possible routes. The drawing was made using the programming language *Processing*, `https://processing.org`.

Challenge 1.2. *How many routes to the result are there in adding up 9 vectors? Or n vectors? Can you describe which of them have been selected for Figure 1.6?*

1.1.2 Scalar multiplication

This corresponds to moving λ times as far, for some real number λ (known as the *scalar*). For scalars, we typically use lower case Greek letters.

We say that the resulting vector is a *scalar multiple* of the original one. The scalar can be negative, in which case we move into the opposite direction. Algebraically, *scalar multiplication* multiplies each coordinate of the vector with the scalar; see Figure1.7.



$$3\begin{bmatrix}2\\1\end{bmatrix}=\begin{bmatrix}6\\3\end{bmatrix} \qquad (-2)\begin{bmatrix}2\\1\end{bmatrix}=\begin{bmatrix}-4\\-2\end{bmatrix}$$

Figure 1.7: Scalar multiplication

Definition 1.3 (Scalar multiplication)**.** *Let*

$$\mathbf{v}=\begin{bmatrix}v_1\\v_2\\\vdots\\v_m\end{bmatrix}\in\mathbb{R}^m,\ \lambda\in\mathbb{R}.\ \textit{The vector } \lambda\mathbf{v}:=\begin{bmatrix}\lambda v_1\\\lambda v_2\\\vdots\\\lambda v_m\end{bmatrix}\in\mathbb{R}^m \textit{ is a } \text{scalar multiple } \textit{of } \mathbf{v}.$$

Note that the zero vector $\mathbf{0}$ is a scalar multiple of every vector, obtained by choosing scalar $\lambda=0$.

1.1.3 Linear combinations

This operation combines vector addition and scalar multiplication.

Definition 1.4 (Linear combination)**.** *Let* $\mathbf{v},\mathbf{w}\in\mathbb{R}^m$, $\lambda,\mu\in\mathbb{R}$. *The vector*

$$\lambda\mathbf{v}+\mu\mathbf{w}\in\mathbb{R}^m$$

is a linear combination *of* $\mathbf{v}$ *and* $\mathbf{w}$. *In general, if* $\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_n\in\mathbb{R}^m$ *and* $\lambda_1,\lambda_2,\ldots,\lambda_n\in\mathbb{R}$, *then*

$$\lambda_1\mathbf{v}_1+\lambda_2\mathbf{v}_2+\cdots+\lambda_n\mathbf{v}_n$$

is a linear combination *of* $\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_n$.

Table 1.1 gives three linear combinations of two specific vectors $\mathbf{v}$ and $\mathbf{w}$. What are all the linear combinations of $\mathbf{v}$ and $\mathbf{w}$ that can we get in this way? Here is the answer:

$$\mathbf{v} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} 3 \\ -1 \end{bmatrix} :$$

λ	μ	$\lambda\mathbf{v}$	$\mu\mathbf{w}$	$\lambda\mathbf{v} + \mu\mathbf{w}$
-3	2	$\begin{bmatrix} -6 \\ -9 \end{bmatrix}$	$\begin{bmatrix} 6 \\ -2 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -11 \end{bmatrix}$
1	-1	$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$	$\begin{bmatrix} -3 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 4 \end{bmatrix}$
3	0	$\begin{bmatrix} 6 \\ 9 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 6 \\ 9 \end{bmatrix}$

Table 1.1: Three linear combinations of two vectors $\mathbf{v}, \mathbf{w}$

Fact 1.5. *Every vector in $\mathbb{R}^2$ is a linear combination of the two vectors*

$$\mathbf{v} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}.$$

Proof. Let

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

be an arbitrary vector in $\mathbb{R}^2$. We need to show that we can find scalars $\lambda, \mu \in \mathbb{R}$ such that

$$\lambda \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \mu \begin{bmatrix} 3 \\ -1 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

Considering the rules of vector addition and scalar multiplication, this vector equation is actually made up of two "normal" equations, one for each coordinate:

$$\begin{aligned} 2\lambda + 3\mu &= u_1, \\ 3\lambda - 1\mu &= u_2. \end{aligned}$$

This is a system of two linear equations in two variables λ and μ, and the rest is therefore a highschool job.

Well, almost: What you may find unusual here is that the system also contains the variables u_1, u_2. But while the variables λ and μ stand for the unknown numbers that we want to compute, u_1 and u_2 stand for known numbers, the entries of our target vector $\mathbf{u}$. By solving this system of equations (the solution will depend on u_1 and u_2), we therefore solve it for all possible values of u_1 and u_2 at the same time. And this was exactly the point, since we want to make the argument for every possible vector $\mathbf{u}$. This is what differentiates a proof from a calculation: a proof makes one argument for many (possibly infinitely many) situations, while a calculation just handles one situation.

After this digression, let's solve the system: Multiplying the second equation by 3 and adding it to the first one cancels the variable μ:

$$\begin{array}{rcrcrcr} 2\lambda & + & 3\mu & = & u_1 & & \\ 9\lambda & - & 3\mu & = & & & 3u_2 \\ \hline 11\lambda & & & = & u_1 & + & 3u_2 \end{array}$$

This gives

$$\lambda = \frac{u_1 + 3u_2}{11}.$$

To get μ, we isolate μ in one of the equations (let's take the first one) and substitute the value of λ that we just got:

$$3\mu = u_1 - 2\lambda = u_1 - \frac{2u_1 + 6u_2}{11} = \frac{11u_1 - (2u_1 + 6u_2)}{11} = \frac{9u_1 - 6u_2}{11}.$$

Dividing by 3 yields

$$\mu = \frac{3u_1 - 2u_2}{11}.$$

So λ and μ have been found for all possible values of u_1 and u_2 which completes the proof. □

It's always good to double check this on examples. Let's look at the three linear combinations that we have previously computed in Table 1.1 and see whether the formulas for λ and μ give us back the correct scalars. Table 1.2 shows that they do.

$$\mathbf{v} = \begin{bmatrix}2\\3\end{bmatrix}, \mathbf{w} = \begin{bmatrix}3\\-1\end{bmatrix}:$$

λ	μ	$\lambda\mathbf{v} + \mu\mathbf{w} = \mathbf{u}$	u_1	u_2	$\frac{u_1+3u_2}{11}$	$\frac{3u_1-2u_2}{11}$
-3	2	$\begin{bmatrix}0\\-11\end{bmatrix}$	0	-11	-3	2
1	-1	$\begin{bmatrix}-1\\4\end{bmatrix}$	-1	4	1	-1
3	0	$\begin{bmatrix}6\\9\end{bmatrix}$	6	9	3	0

Table 1.2: Checking the formulas from the proof of Fact 1.5 on three vectors

We can also understand this proof geometrically. The two equations $2\lambda + 3\mu = u_1$ and $3\lambda - 1\mu = u_2$ can be drawn as lines in the $\lambda\mu$-plane, and their point of intersection is the desired solution to both equations. For $u_1 = -1, u_2 = 4$ (middle row of Table 1.2), the two lines are drawn in Figure 1.8 (left). Unsurprisingly, their intersection is at $(\lambda, \mu) = (1, -1)$.

The crucial observation is that for other values of u_1, u_2, the lines are different but still parallel to the lines for $u_1 = -1, u_2 = 4$; see Figure 1.8 (right). So there is always an intersection, meaning that we find values λ, μ for every vector $\mathbf{u}$. This was the "row picture" behind the proof.

There is also a "column picture", see Figure 1.9. The two vectors $\mathbf{v}$ and $\mathbf{w}$ define axes of a skewed coordinate system (solid lines). Given a target vector $\mathbf{u}$, we make shifted copies of both axes such that they cross in $\mathbf{u}$ (dashed lines). Between the two pairs of axes, a parallelogram emerges, and this is exactly the one that expresses $\mathbf{u}$ as a sum of scaled versions of $\mathbf{v}$ and $\mathbf{w}$, the gray arrows in Figure 1.9 (right).

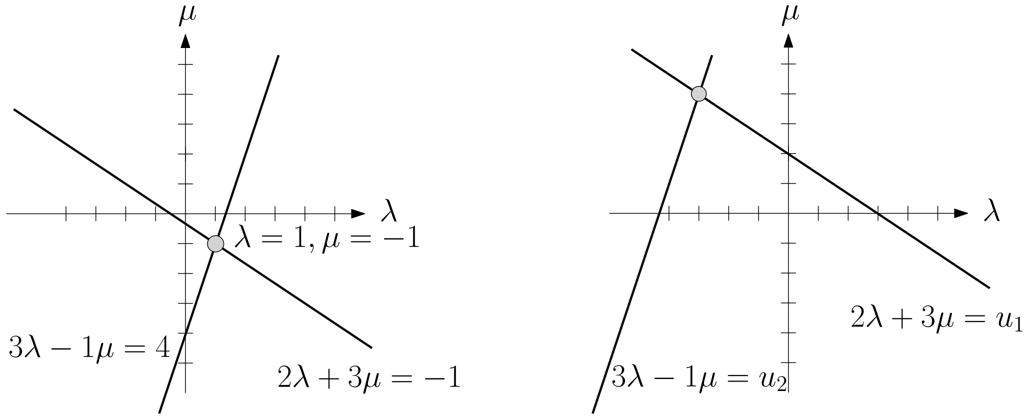


Figure 1.8: "Row picture" behind the proof of Fact 1.5, based on the rows of the equation system. We find the target scalars by intersecting two lines in the $\lambda\mu$-plane.

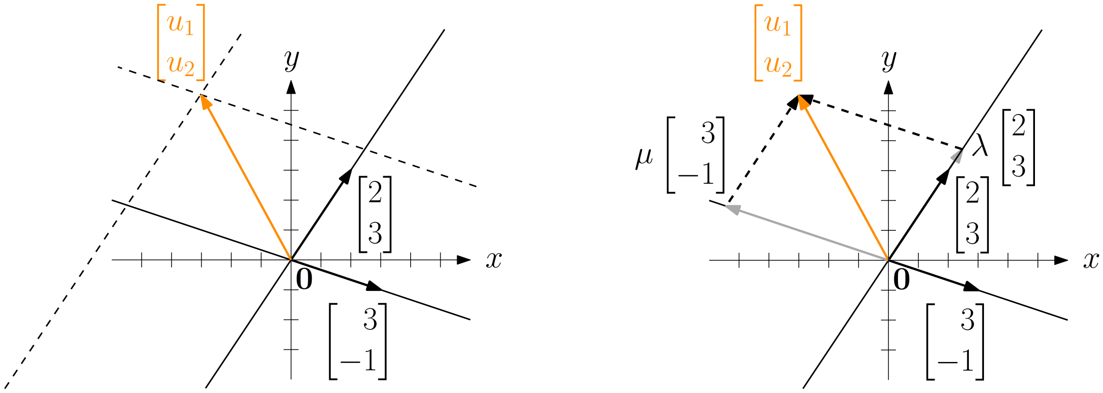

$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$
y
$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$
x
$\mathbf{0}$
$\begin{bmatrix} 3 \\ -1 \end{bmatrix}$
$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$
y
$\mu \begin{bmatrix} 3 \\ -1 \end{bmatrix}$
$\lambda \begin{bmatrix} 2 \\ 3 \end{bmatrix}$
$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$
x
$\mathbf{0}$
$\begin{bmatrix} 3 \\ -1 \end{bmatrix}$

Figure 1.9: "Column picture" behind the proof of Fact 1.5, based on the columns of the equation system: We find the target scalars as coordinates in a skewed coordinate system.

Fact 1.5 may be wrong for other vectors. As an example, consider

$$\mathbf{v} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}.$$

Here, $\mathbf{w}$ is a scalar multiple of $\mathbf{v}$, so every linear combination of the two is just another scalar multiple of $\mathbf{v}$. Therefore, the linear combinations form a line through $\mathbf{v}$, and a vector not on that line cannot be obtained as a linear combination.

Challenge 1.6. *Try to prove a version of Fact 1.5 where*

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

are arbitrary vectors. Where does the proof fail if the two vectors are scalar multiples of each other? What goes wrong in the row and column pictures in this case? And why does the proof indeed succeed when the two vectors are not scalar multiples of each other?

1.1.4 Affine, conic, and convex combinations

In many applications, we are not interested in all linear combinations of the given vectors. The following three types of special linear combinations are of particular importance.

Definition 1.7 (Affine, conic, convex combination). *A linear combination $\lambda_1\mathbf{v}_1+\lambda_2\mathbf{v}_2+\cdots+\lambda_n\mathbf{v}_n$ of vectors $\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_n$ is called*

(i) an affine combination *if $\lambda_1+\lambda_2+\cdots+\lambda_n=1$,*

(ii) a conic combination *if $\lambda_j\geq 0$ for $j=1,2,\ldots,n$, and*

(iii) a convex combination *if it is both an affine and a conic combination.*

We will illustrate these notions with linear combinations $\lambda\mathbf{v}+\mu\mathbf{w}$ of two vectors $\mathbf{v},\mathbf{w}\in\mathbb{R}^2$ that are not scalar multiples of each other. By Challenge 1.6, the set of linear combinations of such vectors is the whole plane $\mathbb{R}^2$; see Figure 1.10 (i). What is the set of affine combinations? If $\lambda+\mu=1$, we can write

$$\lambda\mathbf{v}+\mu\mathbf{w}=\lambda\mathbf{v}+\underbrace{(1-\lambda)}_{\mu}\mathbf{w}=\mathbf{w}+\lambda(\mathbf{v}-\mathbf{w}).$$

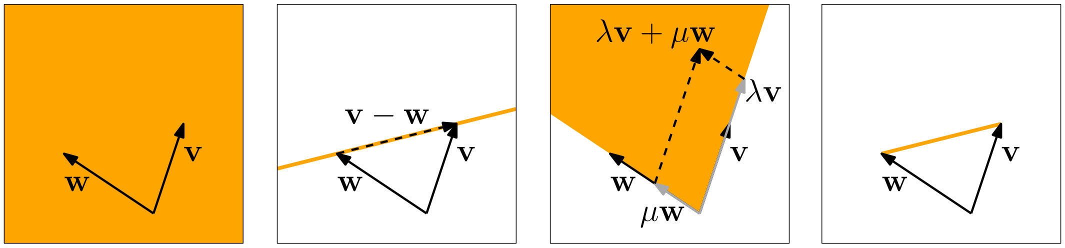
Hence, the set of all affine combinations is the set

$$\{\mathbf{w}+\lambda(\mathbf{v}-\mathbf{w}):\lambda\in\mathbb{R}\}.$$

This is the line through $\mathbf{v}$ and $\mathbf{w}$ (interpreted as points; see Figure 1.3). Plugging in $\lambda=0$ gives $\mathbf{w}$, and for $\lambda=1$, we obtain $\mathbf{v}$. Values of λ between 0 and 1 lead to points in between. For $\lambda<0$ we end up left of $\mathbf{w}$, and for $\lambda>1$, we will be right of $\mathbf{v}$. Generally, any point on the line is obtained by the rule "go to $\mathbf{w}$, then make a step parallel to $\mathbf{v}-\mathbf{w}$!". See Figure 1.10 (ii).

In a conic combination, the parallelogram for adding $\lambda\mathbf{v}$ and $\mu\mathbf{w}$ is always in the angle between $\mathbf{v}$ and $\mathbf{w}$, and by varying λ and μ, we can get every vector in this angle, officially called the *cone* spanned by $\mathbf{v}$ and $\mathbf{w}$; see Figure 1.10 (iii). Finally, the convex combinations are by definition all points on the line through $\mathbf{v}$ and $\mathbf{w}$ that are also in the cone spanned by $\mathbf{v}$ and $\mathbf{w}$. This set is the *line segment* spanned by $\mathbf{v}$ and $\mathbf{w}$, see Figure 1.10 (iv).

Challenge 1.8. *Understand the affine, conic and convex combinations of* three *vectors in $\mathbb{R}^2$!*



linear (i) affine (ii) conic (iii) convex

combinations of two vectors

Figure 1.10: Two vectors and their combinations

1.1.5 Defining the dots: sequences, sums, sets, and vectors

The dots notations as used in

$$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \quad \text{and} \quad \lambda_1\mathbf{v}_1 + \lambda_2\mathbf{v}_2 + \cdots + \lambda_n\mathbf{v}_n \quad \text{and} \quad \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

are fine, but it is never too early to get to know the precise mathematical notations. This will also help in really understanding the dots notations. Formally, when we write $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, we mean the sequence $(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$ of n vectors; we typically omit the surrounding brackets if we don't need the sequence itself as a mathematical object. There is a more concise way of writing the sequence:

$$(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n) = (\mathbf{v}_j)_{j=1}^n.$$

Similarly, sums have a mathematical notation:

$$\lambda_1\mathbf{v}_1 + \lambda_2\mathbf{v}_2 + \cdots + \lambda_n\mathbf{v}_n = \sum_{j=1}^n \lambda_j\mathbf{v}_j.$$

Here, j is called the *summation index*.

The benefit of this notation becomes apparent when we discuss some special cases. What if $n = 2$? It's not entirely clear how to interpret $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ in this case. Does $\mathbf{v}_2$ now appear twice? And if $n = 1$? Then there is not even a vector $\mathbf{v}_2$, so why is it mentioned?

In starting with $\mathbf{v}_1, \mathbf{v}_2, \ldots$, we want to indicate a pattern that continues until $\mathbf{v}_n$, and we therefore mean $(\mathbf{v}_1, \mathbf{v}_2)$ if $n = 2$ and $(\mathbf{v}_1)$ if $n = 1$. Still, it's potentially confusing to always mention three vectors $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_n$, even if there are less. In contrast, the notation

$$(\mathbf{v}_j)_{j=1}^n$$

has a clear definition: it's the sequence of all vectors $\mathbf{v}_j$ where j goes through the range from 1 to n in increasing order. This range contains the integers j that satisfy $1 \leq j \leq n$. This naturally covers the cases $n = 1$ and $n = 2$. For the sum, it's the same:

$$\sum_{j=1}^{n} \lambda_j \mathbf{v}_j$$

is obtained by summing up all $\lambda_j \mathbf{v}_j$ for which j goes through the range from 1 to n.

This brings us to the last special case: what if $n = 0$, so there are no vectors? The mathematical notation handles this elegantly: as the range of integers from 1 to 0 is empty, we have

$$(\mathbf{v}_j)_{j=1}^{0} = (),$$

the empty sequence. And

$$\sum_{j=1}^{0} \lambda_j \mathbf{v}_j = \mathbf{0},$$

the zero vector. Indeed, if you add up vectors, you start from $\mathbf{0}$ and then add one vector at a time. But if there are no vectors to add, you remain stuck at $\mathbf{0}$. This is like a scale that shows a weight of 0 before you put anything on it. As a consequence, $\mathbf{0}$ is a linear combination of *every* sequence of vectors, even the empty one.

Having made it clear that for $n = 0$, $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ is the empty sequence and $\lambda_1\mathbf{v}_1 + \lambda_2\mathbf{v}_2 + \cdots + \lambda_n\mathbf{v}_n = \mathbf{0}$, there is also no harm if we keep using the dot notations. For novices, they may be easier to read than $(\mathbf{v}_j)_{j=1}^{n}$ and $\sum_{j=1}^{n} \lambda_j \mathbf{v}_j$. But as in particular the sum notation is standard throughout many sources, it is also good to get used to it early.

When we reorder a sequence of vectors, we get the same linear combinations, because vector addition is commutative. Also, if a vector appears more than once in the sequence, we can omit its duplicates and still get the same linear combinations. This means, all that matters is the *set* of vectors involved in the sequence $(\mathbf{v}_j)_{j=1}^{n}$. We will write this set as

$$\{\mathbf{v}_j : j \in [n]\}.$$

Here, $[n]$ is an unambiguous notation for the the range from 1 to n, considered as a set. Within a set, the order does not matter, so we prefer the notation $[n]$ over $\{1, 2, \ldots, n\}$. But the latter notation and also $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ or $\{\mathbf{v}_j : j = 1, 2, \ldots, n\}$ for the set of vectors are perfectly fine as well. The important thing is that there is no order of vectors in the set, and every vector only appears once. We can write $\{\mathbf{v}_1, \mathbf{v}_1, \mathbf{v}_2\}$, but this is the same as $\{\mathbf{v}_1, \mathbf{v}_2\}$ or $\{\mathbf{v}_2, \mathbf{v}_1\}$.

Finally, the vertical dots: a vector

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

can more concisely be written as $[v_i]_{i=1}^m$. This is very similar to the sequence notation, and indeed, a vector as an element of $\mathbb{R}^m$ is formally just a sequence of m real numbers. Using square brackets instead of normal brackets indicates that we mean a column vector. This notation for example allows us to write down the vector $[i^2]_{i=1}^5$. What does this mean? The expression in brackets tells us what the i-th entry of the vector should be, and the range after the bracket tells us which range of i's we are considering. Hence,

$$[i^2]_{i=1}^5 = \begin{bmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \end{bmatrix}.$$

Similarly, $[0]_{i=1}^6$ is the 6-dimensional zero vector. Using this notation, we could also define the sum of two vectors more "efficiently" than Definition 1.1 does it, namely as follows:

$$[v_i]_{i=1}^m + [w_i]_{i=1}^m := [v_i + w_i]_{i=1}^m.$$

We can still use the vertical dots notation as it is more readable, despite needing more space. But now that we know what it precisely means, we for example understand that for $m = 1$,

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = [v_1] \in \mathbb{R}^1.$$

What about the case $m = 0$? Are there vectors with no entries? Formally, they should be elements of $\mathbb{R}^0$. Actually, $\mathbb{R}^0$ contains exactly one element, namely the empty sequence $()$ of real numbers. We can rightfully consider this as the 0-dimensional zero vector. But talking about this case in any further detail would get us into nerd territory.

1.2 Scalar products, lengths and angles

The scalar product of two vectors is a number that lets us measure the length of a vector and the angle between two vectors. Scalar products naturally appear all over linear algebra and have many practical applications. Scalar products are also at the heart of two important inequalities, the Cauchy-Schwarz inequality, and the triangle inequality.

Given how vector addition works (Section 1.1.1), you might expect vector multiplication to work like this:

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \cdot 3 \\ 2 \cdot 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \end{bmatrix}.$$

This is known as the *Hadamard product*, but in the range of linear algebra products, it only occupies a niche. Among the topsellers, we find the scalar product whose result is not a vector, but a number (or a scalar, in linear algebra jargon; this is where the name comes from).

1.2.1 Scalar product

The *scalar product* of two vectors is obtained by multiplying corresponding coordinates, and adding up the products:

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

Definition 1.9 (Scalar product)**.** *Let*

$$\mathbf{v} = \begin{bmatrix}v_1\\v_2\\\vdots\\v_m\end{bmatrix}, \mathbf{w} = \begin{bmatrix}w_1\\w_2\\\vdots\\w_m\end{bmatrix} \in \mathbb{R}^m.$$

The scalar product of $\mathbf{v}$ *and* $\mathbf{w}$ *is the number*

$$\mathbf{v} \cdot \mathbf{w} := v_1w_1 + v_2w_2 + \cdots + v_mw_m = \sum_{i=1}^{m} v_iw_i.$$

From this definition, we can directly derive some rules that are frequently needed in computing with scalar products. We summarize these in an *observation*. This is a statement whose proof is simple and straightforward enough to be omitted.

Observation 1.10. *Let* $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^m$ *be vectors and* $\lambda \in \mathbb{R}$ *a scalar. Then*

(i) $\mathbf{v} \cdot \mathbf{w} = \mathbf{w} \cdot \mathbf{v}$*;* *(*symmetry*)*

(ii) $(\lambda\mathbf{v}) \cdot \mathbf{w} = \lambda(\mathbf{v} \cdot \mathbf{w}) = \mathbf{v} \cdot (\lambda\mathbf{w})$*;* *(*taking out scalars*)*

(iii) $\mathbf{u} \cdot (\mathbf{v} + \mathbf{w}) = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \cdot \mathbf{w}$ *and* $(\mathbf{u} + \mathbf{v}) \cdot \mathbf{w} = \mathbf{u} \cdot \mathbf{w} + \mathbf{v} \cdot \mathbf{w}$*;* *(*distributivity*)*

(iv) $\mathbf{v} \cdot \mathbf{v} \geq 0$*, with equality exactly if* $\mathbf{v} = \mathbf{0}$*.* *(*positive-definiteness*)*

While (i) and (iv) are quite obvious from the definition of the scalar product, (ii) and (iii) need some straightforward calculations as a proof. To show what we mean by "straightforward", we provide the calculations for the first part of (iii). These use distributivity in $\mathbb{R}$:

$$\mathbf{u} \cdot (\mathbf{v} + \mathbf{w}) = \sum_{i=1}^{m} u_i(v_i + w_i) = \sum_{i=1}^{m}(u_iv_i + u_iw_i) = \sum_{i=1}^{m} u_iv_i + \sum_{i=1}^{m} u_iw_i = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \cdot \mathbf{w}.$$

Properties (ii) and (iii) together are known as *linearity in both arguments*.

1.2.2 Euclidean norm

The *Euclidean norm* of a vector $\mathbf{v}$ is obtained by taking the square root of the scalar product with itself. We can do this, since this scalar product is nonnegative by Observation 1.10 (iv).

Definition 1.11 (Euclidean norm). *Let* $\mathbf{v} \in \mathbb{R}^m$. *The Euclidean norm of* $\mathbf{v}$ *is the number*

$$\|\mathbf{v}\| := \sqrt{\mathbf{v} \cdot \mathbf{v}}.$$

Some sources use $|\mathbf{v}|$ for the norm, but we reserve this notation for the *absolute value* of a number; for example, $|3| = |-3| = 3$.

You can think of the Euclidean norm as defining the length of a vector. You may argue that we don't need to define this, since a vector already has a length (just measure how long the arrow is). But this is true only in $\mathbb{R}^2$ and $\mathbb{R}^3$ where we can draw vectors as arrows. In higher dimensions, it is not a priori clear how to measure the length of a vector $\mathbf{v}$. But now we know: compute its Euclidean norm $\|\mathbf{v}\|$!

In $\mathbb{R}^2$ and $\mathbb{R}^3$, the Euclidean norm indeed measures the length of the arrow; so we're not really inventing a new concept of length here, we simply extend a familiar concept to higher dimensions. To see this, we first expand the scalar product to obtain the following formula for the Euclidean norm:

$$\left\| \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \right\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_m^2} = \sqrt{\sum_{i=1}^{m} v_i^2}.$$

For example,

$$\left\| \begin{bmatrix} -4 \\ 2 \end{bmatrix} \right\| = \sqrt{(-4)^2 + 2^2} = \sqrt{20}, \qquad \left\| \begin{bmatrix} -4 \\ 2 \\ 3 \end{bmatrix} \right\| = \sqrt{(-4)^2 + 2^2 + 3^2} = \sqrt{29}.$$

Let's first look at the situation in $\mathbb{R}^2$ where Figure 1.11 is the key.

The vector

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

is the hypotenuse of a right-angled triangle whose legs have lengths $|v_1|$ and $|v_2|$, see Figure 1.11. Hence, using the *Pythagorean theorem*, the squared length of $\mathbf{v}$ is

$$|v_1|^2 + |v_2|^2 = v_1^2 + v_2^2 = \|\mathbf{v}\|^2,$$

and "length of $\mathbf{v}$ equals $\|\mathbf{v}\|$ is obtained by taking square roots. Building on this, Figure 1.12 deals with the situation in $\mathbb{R}^3$.

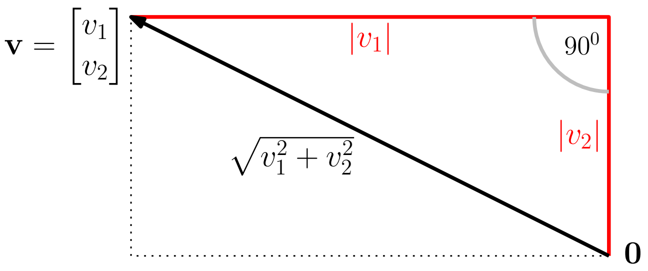
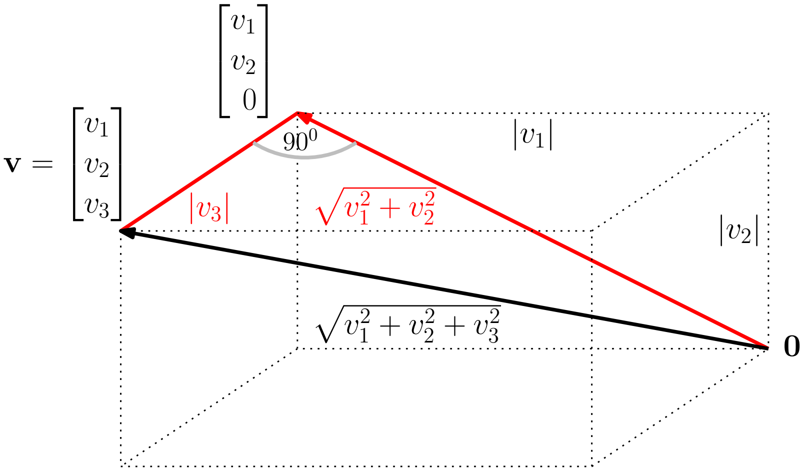


Figure 1.11: The Euclidean norm measures the length of a vector in $\mathbb{R}^2$.



$\begin{bmatrix} v_1 \\ v_2 \\ 0 \end{bmatrix}$
$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$
90^0
$|v_1|$
$|v_3|$
$\sqrt{v_1^2 + v_2^2}$
$|v_2|$
$\sqrt{v_1^2 + v_2^2 + v_3^2}$
$\mathbf{0}$

Figure 1.12: The Euclidean norm measures the length of a vector in $\mathbb{R}^3$.

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

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

is the hypotenuse of a right-angled triangle whose legs have lengths $\sqrt{v_1^2 + v_2^2}$ (as just computed) and $|v_3|$; see Figure 1.12. Thus, Pythagoras tells us that the squared length of $\mathbf{v}$ is

$$\sqrt{v_1^2 + v_2^2}^2 + |v_3|^2 = v_1^2 + v_2^2 + v_3^2 = \|\mathbf{v}\|^2.$$

Unit vectors. A *unit vector* is a vector $\mathbf{u}$ such that $\|\mathbf{u}\| = 1$. In $\mathbb{R}^2$, the unit vectors lie on the *unit circle* with center $\mathbf{0}$ and radius 1; see Figure 1.13.

Using Definition 1.11 of the Euclidean norm, and taking out scalars (Observation 1.10 (ii)), it is easy to see that for every vector $\mathbf{v} \neq \mathbf{0}$, the vector

$$\frac{\mathbf{v}}{\|\mathbf{v}\|}$$

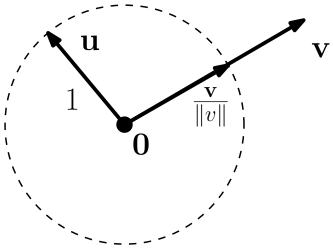


Figure 1.13: A unit vector $\mathbf{u}$ on the unit circle. For every nonzero vector $\mathbf{v}$, the scaled vector $\mathbf{v}/\|\mathbf{v}\|$ is a unit vector.

is a unit vector, where *scalar division* is just scalar multiplication with the reciprocal:

$$\frac{\mathbf{v}}{\|\mathbf{v}\|} := \frac{1}{\|\mathbf{v}\|}\mathbf{v}.$$

In $\mathbb{R}^m$, there are m *standard unit vectors*. These are the ones that have one coordinate equal to 1 and all others equal to 0. We use the notation $\mathbf{e}_i$ for the standard unit vector that has the 1 at the i-th coordinate:

$$\mathbb{R}^3 : \mathbf{e}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix}0\\0\\1\end{bmatrix} \qquad \Bigg| \qquad \mathbb{R}^m : \mathbf{e}_i = \begin{bmatrix}0\\\vdots\\1\\\vdots\\0\end{bmatrix} \leftarrow \text{coordinate } i$$

In $\mathbb{R}^2$, the two standard unit vectors are the ones in the directions of x- and y-axis. In $\mathbb{R}^3$, $\mathbf{e}_3$ goes along the z-axis; see Figure 1.14.

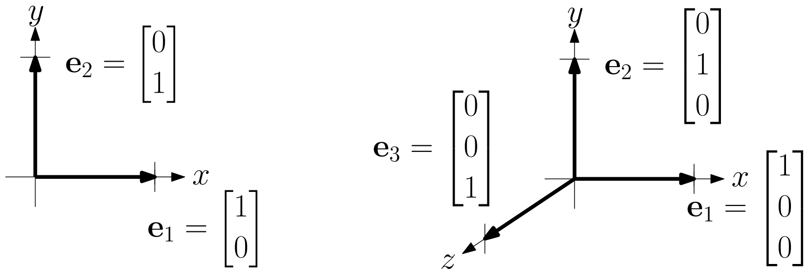

y
$\mathbf{e}_2 = \begin{bmatrix}0\\1\end{bmatrix}$
x
$\mathbf{e}_1 = \begin{bmatrix}1\\0\end{bmatrix}$
y
$\mathbf{e}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}$
$\mathbf{e}_3 = \begin{bmatrix}0\\0\\1\end{bmatrix}$
x
$\mathbf{e}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}$
z

Figure 1.14: The standard unit vectors in $\mathbb{R}^2$ and $\mathbb{R}^3$

Other norms. To stress the point that the length of a vector is nothing God-given, we remark that there are many other norms for vectors that make sense and are being used.

Here are two examples, the *1-norm* and the ∞*-norm*:

$$\|\mathbf{v}\|_1 := \sum_{i=1}^{m} |v_i| \quad \text{(1-norm)}$$

and

$$\|\mathbf{v}\|_\infty := \max_{i=1}^{m} |v_i| \quad (\infty\text{-norm}).$$

In these norms, the "unit circles" look different, see Figure 1.15. The vectors in $\mathbb{R}^2$ that have length 1 according to the 1-norm form a "diamond"; under the ∞-norm, we get a square. The standard unit vectors are also unit vectors under the 1-norm and the ∞-norm.

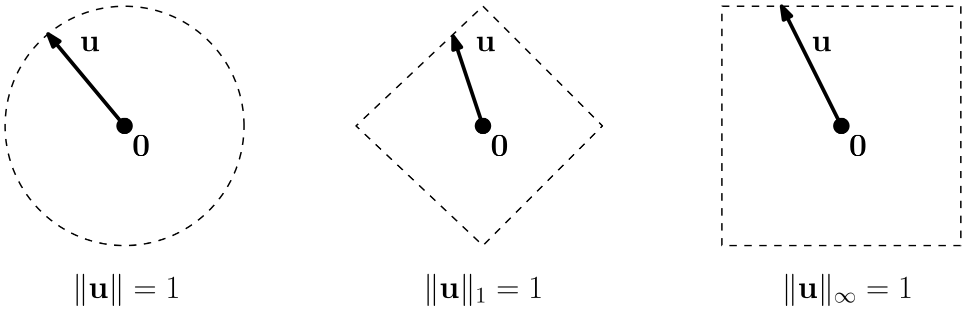


Figure 1.15: Unit vectors in the Euclidean norm (left), 1-norm (middle), ∞-norm (right)

All three norms are special cases of p-norms, where $p \geq 1$ can be any real number:

$$\|\mathbf{v}\|_p = \sqrt[p]{\sum_{i=1}^{m} |v_i|^p}.$$

We see that the Euclidean norm is actually the 2-norm, but we still write it as $\|\mathbf{v}\|$, not $\|\mathbf{v}\|_2$, since for us, it is the "standard" norm. The ∞-norm is an abuse of notation, since ∞ is not a real number. But as "p goes to infinity", the largest coordinate of $\mathbf{v}$ in absolute value is all that matters. In formulas,

$$\lim_{p\to\infty} \|\mathbf{v}\|_p = \|\mathbf{v}\|_\infty.$$

1.2.3 Cauchy-Schwarz inequality

As innocent as it looks, the importance of this inequality cannot be overestimated.

Lemma 1.12 (Cauchy-Schwarz inequality). *For any two vectors* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^m$*,*

$$|\mathbf{v} \cdot \mathbf{w}| \leq \|\mathbf{v}\|\|\mathbf{w}\|.$$

Moreover, equality holds exactly if one vector is a scalar multiple of the other.

Generally, a *lemma* is a helper statement that may not be very interesting on its own; but the more it actually helps, the more important it becomes. In this sense, a lemma is like a Swiss army knife, where the Cauchy-Schwarz inequality is a highly multifunctional one.

Proof. We first consider the case where $\mathbf{v}$ and $\mathbf{w}$ are unit vectors, so $\|\mathbf{v}\| = \|\mathbf{w}\| = 1$. Using Observation 1.10 and Definition 1.11 of the Euclidean norm, we compute

$$\begin{aligned}
0 \le (\mathbf{v}-\mathbf{w})\cdot(\mathbf{v}-\mathbf{w}) &= \underbrace{\mathbf{v}\cdot\mathbf{v}}_{\|\mathbf{v}\|^2} + \underbrace{\mathbf{w}\cdot\mathbf{w}}_{\|\mathbf{w}\|^2} - 2\mathbf{v}\cdot\mathbf{w} = 2 - 2\mathbf{v}\cdot\mathbf{w} \quad \Rightarrow \quad \mathbf{v}\cdot\mathbf{w} \le 1,\\
0 \le (\mathbf{v}+\mathbf{w})\cdot(\mathbf{v}+\mathbf{w}) &= \overbrace{\mathbf{v}\cdot\mathbf{v}} + \overbrace{\mathbf{w}\cdot\mathbf{w}} + 2\mathbf{v}\cdot\mathbf{w} = 2 + 2\mathbf{v}\cdot\mathbf{w} \quad \Rightarrow \quad \mathbf{v}\cdot\mathbf{w} \ge -1.
\end{aligned}$$

Summarized as $|\mathbf{v}\cdot\mathbf{w}| \le 1$, this proves the Cauchy-Schwarz inequality for unit vectors.

Now suppose $\mathbf{v}$ and $\mathbf{w}$ are arbitrary vectors. If one of them is $\mathbf{0}$, the inequality holds (both sides are 0). If $\mathbf{v} \ne \mathbf{0}$ and $\mathbf{w} \ne \mathbf{0}$, we can apply the previous calculations after scaling $\mathbf{v}$ and $\mathbf{w}$ to unit length. This gives

$$-1 \le \frac{\mathbf{v}}{\|\mathbf{v}\|}\cdot\frac{\mathbf{w}}{\|\mathbf{w}\|} \le 1,$$

Taking out scalars according to Observation 1.10 (ii) and multiplying all three terms with $\|\mathbf{v}\|\|\mathbf{w}\|$ results in

$$-1 \le \frac{\mathbf{v}\cdot\mathbf{w}}{\|\mathbf{v}\|\|\mathbf{w}\|} \le 1 \quad \text{and} \quad -\|\mathbf{v}\|\|\mathbf{w}\| \le \mathbf{v}\cdot\mathbf{w} \le \|\mathbf{v}\|\|\mathbf{w}\|.$$

This can be summarized as $|\mathbf{v}\cdot\mathbf{w}| \le \|\mathbf{v}\|\|\mathbf{w}\|$ which proves the Cauchy-Schwarz inequality in general.

For the "Moreover" part, we need to understand under which conditions equality holds. Going back to the calculations with the unit vectors, we see that the conditions are precisely $0 = (\mathbf{v}-\mathbf{w})(\mathbf{v}-\mathbf{w})$ or $0 = (\mathbf{v}+\mathbf{w})(\mathbf{v}+\mathbf{w})$. By positive-definiteness of the scalar product (Observation 1.10(iv)), this translates to $\mathbf{v}-\mathbf{w} = \mathbf{0}$ or $\mathbf{v}+\mathbf{w} = \mathbf{0}$.

In other words, the two unit vectors are either the same or opposite vectors. For the unit vectors $\mathbf{v}/\|\mathbf{v}\|$ and $\mathbf{w}/\|\mathbf{w}\|$, it is easy to see that this is the case exactly if $\mathbf{v}$ and $\mathbf{w}$ are scalar multiples of each other. If equality is due to one of $\mathbf{v}$ and $\mathbf{w}$ being $\mathbf{0}$, it's still true that one vector (namely $\mathbf{0}$) is a scalar multiple of the other one. □

Here is an application of the Cauchy-Schwarz inequality. Imagine that you have m squares with total area A, and you put them next to each other as in Figure 1.16. How much horizontal space do you need at most?

To solve this, let $v_1, v_2, \ldots, v_m$ be the side lengths of the squares. Then the horizontal space needed is

$$\sum_{i=1}^{m} v_i.$$

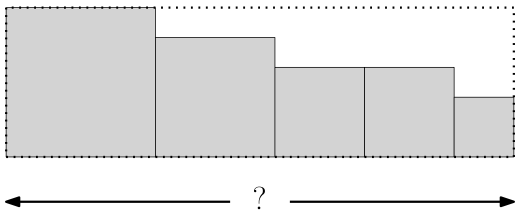


Figure 1.16: Horizontal alignment of m squares with total area A

Let $\mathbf{v} \in \mathbb{R}^m$ be the vector with coordinates $v_1, v_2, \ldots, v_m$. We know that

$$\|\mathbf{v}\|^2 = \sum_{i=1}^{m} v_i^2 = A.$$

Furthermore, let $\mathbf{1} \in \mathbb{R}^m$ be the vector with all coordinates equal to 1. We have $\|\mathbf{1}\| = \sqrt{m}$. By the Cauchy-Schwarz inequality,

$$\sum_{i=1}^{m} v_i = \mathbf{1} \cdot \mathbf{v} \leq \|\mathbf{1}\|\|\mathbf{v}\| = \sqrt{m}\sqrt{A},$$

so we need at most $\sqrt{mA}$ horizontal space. If $\mathbf{v}$ is a scalar multiple of $\mathbf{1}$ (meaning that all squares have the same size), we need exactly $\sqrt{mA}$. Otherwise, we need less.

Exercise 1.13. *For the 1-norm and ∞-norm as defined on page 26, prove that the following inequalities hold for every vector $\mathbf{v} \in \mathbb{R}^m$.*

$$\|\mathbf{v}\| \leq \|\mathbf{v}\|_1 \leq \sqrt{m}\|\mathbf{v}\|$$

and

$$\|\mathbf{v}\|_\infty \leq \|\mathbf{v}\| \leq \sqrt{m}\|\mathbf{v}\|_\infty.$$

1.2.4 Angles

In $\mathbb{R}^2$ and $\mathbb{R}^3$, the angle between two vectors is simply the angle between their arrows; see Figure 1.17.

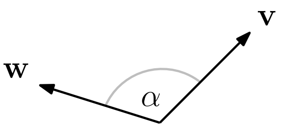


Figure 1.17: The angle α between two vectors $\mathbf{v}$ and $\mathbf{w}$

As with the length, we are looking for a way to define the angle between two vectors also in higher dimensions, in such a way that nothing changes in $\mathbb{R}^2$ and $\mathbb{R}^3$. The Cauchy-Schwarz inequality is the key here.

Definition 1.14 (Angle). *Let $\mathbf{v}, \mathbf{w} \in \mathbb{R}^m$ be two nonzero vectors. The angle between them is the unique α between 0 and π (180 degrees) such that*

$$\cos(\alpha) = \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\|\|\mathbf{w}\|} \in [-1, 1].$$

(Here, the interval $[-1, 1] = \{x \in \mathbb{R} : -1 \leq x \leq 1\}$ *comes from the Cauchy Schwarz inequality, Lemma 1.12). In other words,*

$$\alpha = \arccos\left(\frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\|\|\mathbf{w}\|}\right).$$

Let's check that this coincides with the usual concept of angles in $\mathbb{R}^2$. As the angle does not depend on how long the arrows are, and whether we simultaneously rotate them, we can look at the case where $\mathbf{v} = \mathbf{e}_1$ and $\mathbf{w}$ is another unit vector, with an angle of α between the arrows; see Figure 1.18.

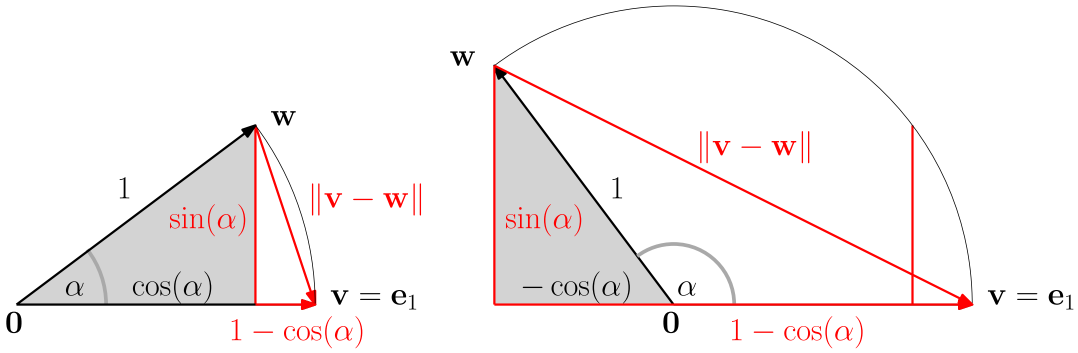


Figure 1.18: The angle α between two unit vectors in $\mathbb{R}^2$. Left: α acute; right: α obtuse

From highschool, we know that the legs of the gray triangle (whose hypotenuse is $\|\mathbf{w}\| = 1$) are $\sin(\alpha)$ and $\cos(\alpha)$ (if α is an acute angle) or $-\cos(\alpha)$ (if α is an obtuse angle). In both cases, the red triangle with hypotenuse $\|\mathbf{v} - \mathbf{w}\|$ therefore has legs $\sin(\alpha)$ and $1 - \cos(\alpha)$. By the Pythagorean theorem,

$$\|\mathbf{v} - \mathbf{w}\|^2 = \sin^2(\alpha) + (1 - \cos(\alpha))^2.$$

Using $\sin^2(\alpha) + \cos^2(\alpha) = 1$, the right-hand side is $2 - 2\cos(\alpha)$. By definition of the Euclidean norm, the left-hand side is $(\mathbf{v} - \mathbf{w}) \cdot (\mathbf{v} - \mathbf{w})$, and we have already argued in the proof of Lemma 1.12 (Cauchy-Schwarz inequality) that this equals $2 - 2\mathbf{v} \cdot \mathbf{w}$. Hence, we have shown

$$2 - 2\mathbf{v} \cdot \mathbf{w} = 2 - 2\cos(\alpha)$$

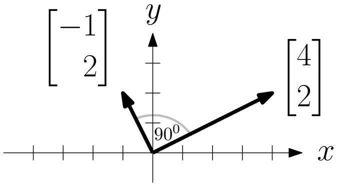
which simplifies to

$$\cos(\alpha) = \mathbf{v} \cdot \mathbf{w}.$$

This indeed agrees with what Definition 1.14 says for unit vectors.

Definition 1.15 (Perpendicular vectors)**.** *Two vectors* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^m$ *are called* perpendicular *(a different term used is* orthogonal*) if* $\mathbf{v} \cdot \mathbf{w} = 0$*; in other words, if the cosine of the angle between them is* 0 *and the angle itself is* 90 *degrees.*

Figure 1.19 gives an example.


$\begin{bmatrix}-1\\2\end{bmatrix}$
y
$\begin{bmatrix}4\\2\end{bmatrix}$
90^0
x

$$\begin{bmatrix}4\\2\end{bmatrix} \cdot \begin{bmatrix}-1\\2\end{bmatrix} = -4 \cdot 1 + 2 \cdot 2 = 0$$

Figure 1.19: Perpendicular vectors: the scalar product equals 0.

1.2.5 Triangle inequality

Lemma 1.16. *Let* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^m$*. Then*

$$\|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|.$$

In $\mathbb{R}^2$, Figure 1.20 shows that this is quite obvious from the parallelogram of vector addition (Figure 1.4).

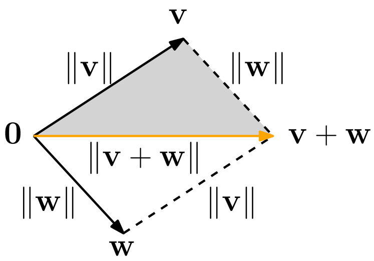


Figure 1.20: The triangle inequality: going from $\mathbf{0}$ directly to $\mathbf{v} + \mathbf{w}$ is shorter than making a detour via $\mathbf{v}$ or $\mathbf{w}$.

In higher dimensions, the following proof shows that the triangle inequality is nothing but Cauchy-Schwarz in a different disguise.

Proof. Since both sides of the inequality are nonnegative, we can instead prove the squared triangle inequality

$$\|\mathbf{v} + \mathbf{w}\|^2 \leq (\|\mathbf{v}\| + \|\mathbf{w}\|)^2$$

and then take square roots on both sides to obtain the triangle inequality. To prove the squared version, we compute

$$\begin{aligned}
\|\mathbf{v}+\mathbf{w}\|^2 &= (\mathbf{v}+\mathbf{w})\cdot(\mathbf{v}+\mathbf{w}) \quad \text{(Definition 1.11) of the Euclidean norm)} \\
&= \mathbf{v}\cdot\mathbf{v}+\mathbf{w}\cdot\mathbf{w}+2\mathbf{v}\cdot\mathbf{w} \quad \text{(Observation 1.10 (iii) on scalar products)} \\
&= \|\mathbf{v}\|^2+\|\mathbf{w}\|^2+2\mathbf{v}\cdot\mathbf{w} \quad \text{(Definition 1.11 of the Euclidean norm)} \\
&\leq \|\mathbf{v}\|^2+\|\mathbf{w}\|^2+2|\mathbf{v}\cdot\mathbf{w}| \\
&\leq \|\mathbf{v}\|^2+\|\mathbf{w}\|^2+2\|\mathbf{v}\|\|\mathbf{w}\| \quad \text{(Cauchy Schwarz inequality, Lemma 1.12)} \\
&= (\|\mathbf{v}\|+\|\mathbf{w}\|)^2.
\end{aligned}$$

□

Exercise 1.17. *Turn this proof around and derive the Cauchy-Schwarz inequality from the (squared) triangle inequality!*

As a consequence, some sources say that the Cauchy-Schwarz inequality is *equivalent* to the triangle inequality; this is another abuse of notation, since any two statements that are both true are logically equivalent, even if they have otherwise nothing to do with each other. What is meant here is that the triangle inequality can easily be proved using the Cauchy-Schwarz inequality (as we did in the proof above), and vice versa, as we ask you to do in Exercise 1.17.

1.3 Linear independence

Linear independence of vectors is probably the single most important concept of linear algebra. A sequence of vectors is called linearly independent if none of the vectors is a linear combination of the others, and linearly dependent otherwise. We provide a number of alternative definitions of linear (in)dependence that illuminate the concept from different angles. We also introduce the span of a sequence of vectors, the set of all their linear combinations. Adding some linear combination as a new vector to the sequence does not change the span.

1.3.1 Definition and examples

Definition 1.18 (Linear (in)dependence)**.** *Vectors* $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ *are* linearly dependent *if at least one of them is a linear combination of the others, i.e. there exists an index* $k \in [n]$ *and scalars* λ_j *such that*

$$\mathbf{v}_k = \sum_{\substack{j=1 \\ j\neq k}}^{n} \lambda_j \mathbf{v}_j.$$

Otherwise, $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ *are* linearly independent*.*

Here, the additional "$j \neq k$" below the sum adds a *condition* to the j's considered in the sum: take only the ones in the given range that satisfy the condition. Hence, the sum is over all j except k, so the equation indeed says that $\mathbf{v}_k$ is a linear combination of the other vectors.

Again, let's do some examples. The two vectors

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

are linearly dependent, because

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 4 \\ 6 \end{bmatrix} \text{ or } \begin{bmatrix} 4 \\ 6 \end{bmatrix} = 2 \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

We also call these two vectors *collinear* because they are are on the same line; see Figure 1.21 (left). In contrast,

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

are linearly independent, as none of them is a linear combination (scalar multiple) of the other one; see Figure 1.21 (right).

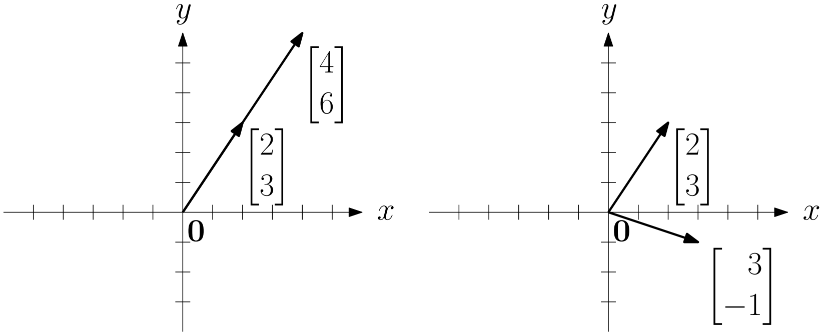

y
$\begin{bmatrix} 4 \\ 6 \end{bmatrix}$
$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$
x
$\mathbf{0}$
y
$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$
x
$\mathbf{0}$
$\begin{bmatrix} 3 \\ -1 \end{bmatrix}$

Figure 1.21: Two collinear vectors (left): their linear combinations form a line; two linearly independent vectors (right)

Let us now consider three vectors in $\mathbb{R}^2$. These are always linearly dependent: If two of them are collinear, one of them is a linear combination of the other one and therefore of *both* other ones (pick scalar 0 for the second other one). Otherwise, each of the vectors is a linear combination of the other two by Challenge 1.6.

What if we have just one vector $\mathbf{v} \in \mathbb{R}^m$? Can $\mathbf{v}$ even be a linear combination of the other vectors when there are no other vectors? Yes, if $\mathbf{v} = \mathbf{0}$, because $\mathbf{0}$ is a linear combination of the empty sequence of vectors (Section 1.1.5). But if $\mathbf{v} \neq \mathbf{0}$, $\mathbf{v}$ is linearly independent.

Generally, when $\mathbf{0}$ is one of the vectors, they are automatically linearly dependent, because $\mathbf{0}$ is always a linear combination of the other ones, even if there are no other ones.

Similarly, when some vector appears twice, the vectors are linearly dependent, because one of the copies is already a linear combination of the other copy.

Finally, what about the empty sequence of vectors? This in turn is linearly independent by Definition 1.18: because $[n] = \emptyset$ in this case ($\emptyset$ is the symbol for the *empty set*), there is no index $k \in [n]$, whatever we may require of it. Table 1.3 summarizes these examples.

linearly independent	linearly dependent
$\begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 3 \\ -1 \end{bmatrix}$	
	$\begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 4 \\ 6 \end{bmatrix}$
	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbb{R}^2$
$\mathbf{v} \neq \mathbf{0}$	
	$\mathbf{v} = \mathbf{0}$
	$\ldots, \mathbf{0}, \ldots$
	$\ldots, \mathbf{v}, \ldots, \mathbf{v}, \ldots$
empty sequence	

Table 1.3: Linear (in)depencence of some sequences of vectors

1.3.2 Alternative definitions

There are two more important alternative definitions of linear dependence. The following lemma provides them. In some sources, (ii) is the "standard" definition of linear dependence.

Lemma 1.19 (Alternative definitions of linear dependence)**.** *Let* $\mathbf{v}_1, \mathbf{v}_2 \ldots, \mathbf{v}_n \in \mathbb{R}^m$*. The following statements are* equivalent *(meaning that they are either all true, or all false).*

(i) At least one of the vectors is a linear combination of the other ones. (This means, the vectors are linearly dependent according to Definition 1.18.)

(ii) There are scalars $\lambda_1, \lambda_2, \ldots, \lambda_n$ *besides* $0, 0, \ldots, 0$ *such that* $\sum_{j=1}^{n} \lambda_j \mathbf{v}_j = \mathbf{0}$*. We also say that* $\mathbf{0}$ *is a* nontrivial linear combination *of the vectors.*

(iii) At least one of the vectors is a linear combination of the previous ones.

For the proof, we apply the basic principles of logic. We first argue that (i) *implies* (ii), meaning that *if* (i) is true, *then* also (ii) is true. Logically, this is written as (i)⇒(ii). Next we prove (ii)⇒(iii) and (iii)⇒(i).

Having done this, we know that (i), (ii) and (ii) are equivalent: either all true or all false, written as (i)⇔(ii)⇔(iii). Indeed, because of the three (circular) implications, it cannot be that one of them is true and another one is false.

In math prose, an equivalence such as (i)⇔(ii) is also written as "(i) *if and only if* (ii)". This summarizes the two implications : "(i) *if* (ii)" writes out (ii)⇒(i), while "(i) *only if* (ii)" excludes the possibilty that (i) is true and (ii) is false. In other words, it writes out the implication (i)⇒(ii).

Proof.

(i)⇒(ii): If at least one of the vectors, $\mathbf{v}_k$ say, is a linear combination of the other vectors, then

$$\mathbf{v}_k = \sum_{\substack{j=1\\ j\neq k}}^{n} \lambda_j \mathbf{v}_j.$$

Defining $\lambda_k = -1$, we get

$$\mathbf{0} = \sum_{j=1}^{n} \lambda_j \mathbf{v}_j.$$

Hence, $\mathbf{0}$ is a nontrivial linear combination of the vectors (it's nontrivial because $\lambda_k \neq 0$).

(ii)⇒(iii): If $\mathbf{0}$ is a nontrivial linear combination of the vectors, then we can write $\mathbf{0}$ in the form

$$\mathbf{0} = \sum_{j=1}^{n} \lambda_j \mathbf{v}_j,$$

where not all λ_j are zero. Let k be the largest index such that $\lambda_k \neq 0$. Then we actually have

$$\mathbf{0} = \sum_{j=1}^{k} \lambda_j \mathbf{v}_j,$$

and this can be solved for $\mathbf{v}_k$, resulting in

$$\mathbf{v}_k = \sum_{j=1}^{k-1} \left(-\frac{\lambda_j}{\lambda_k}\right) \mathbf{v}_j.$$

Hence, at least one vector, namely $\mathbf{v}_k$, is a linear combination of the previous ones.

(iii)⇒(i): If at least one vector is a linear combination of the previous ones, the same vector is also a linear combination of the other ones (use scalar 0 for vectors after it). □

Here are the corresponding alternative definitions of linear *in*dependence. These are simply obtained by taking the opposites of (i)–(iii) in Lemma 1.19: If some statements are either all true or all false, the same holds for their opposites.

We formulate the resulting definitions as a *corollary* which is a result that directly *follows* from (is implied by) a previous one, without the need for a proof (or only a very simple proof such as "take the opposites of all statements!").

Corollary 1.20 (Alternative definitions of linear independence)**.** *Let* $\mathbf{v}_1, \mathbf{v}_2 \ldots, \mathbf{v}_n \in \mathbb{R}^m$*. The following statements are equivalent (meaning that they are either all true, or all false).*

(i) None of the vectors is a linear combination of the other ones. (This means, the vectors are linearly independent according to Definition 1.18.)

(ii) There are no scalars $\lambda_1, \lambda_2, \ldots, \lambda_n$ *besides* $0, 0, \ldots, 0$ *such that* $\sum_{j=1}^{n} \lambda_j \mathbf{v}_j = \mathbf{0}$*. We also say that* $\mathbf{0}$ *can only be written as a* trivial linear combination *of the vectors.*

(iii) None of the vectors is a linear combination of the previous ones.

This has another important consequence: a linear combination of linearly independent vectors can be written as a linear combination *in only one way*.

Lemma 1.21. *Let* $\mathbf{v}_1, \mathbf{v}_2 \ldots, \mathbf{v}_n \in \mathbb{R}^m$ *be linearly independent, and let* $\mathbf{v} = \sum_{j=1}^{n} \lambda_j \mathbf{v}_j = \sum_{j=1}^{n} \mu_j \mathbf{v}_j$ *be two ways of writing* $\mathbf{v}$ *as a linear combination. Then* $\lambda_j = \mu_j$ *for all* $j \in [n]$*.*

Proof. Subtracting the two linear combinations, we get

$$\mathbf{0} = \sum_{j=1}^{n} (\lambda_j - \mu_j) \mathbf{v}_j.$$

Since $\mathbf{0}$ can only be written as a trivial linear combination, we get $\lambda_j - \mu_j = 0$ for all j. □

1.3.3 Span of vectors

The set of all linear combinations of some vectors is important enough to deserve a name.

Definition 1.22 (Span)**.** *Let* $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \in \mathbb{R}^m$*. Their* span *is the set of all linear combinations. In formulas,*

$$\mathbf{Span}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n) := \left\{ \sum_{j=1}^{n} \lambda_j \mathbf{v}_j : \lambda_j \in \mathbb{R} \textit{ for all } j \in [n] \right\}.$$

As an example, let us consider the span of three vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ in $\mathbb{R}^3$. There are three cases; if you have looked into Challenge 1.6, this will not surprise you. Formally, the different cases result from analyzing systems of three linear equations in three variables (which we will not do now).

The span can be a line through the origin (vectors are *collinear*), a plane through the origin (vectors are *coplanar*), or the whole space (vectors are linearly independent). Figure 1.22 presents examples for all three cases.

The attentive reader might have noticed that there is a fourth (but pretty boring) case: if $\mathbf{v}_1 = \mathbf{v}_2 = \mathbf{v}_3 = \mathbf{0}$, then $\mathbf{0}$ is the only linear combination, so the span is a point at the origin. Here are some more observations to illustrate the concept.

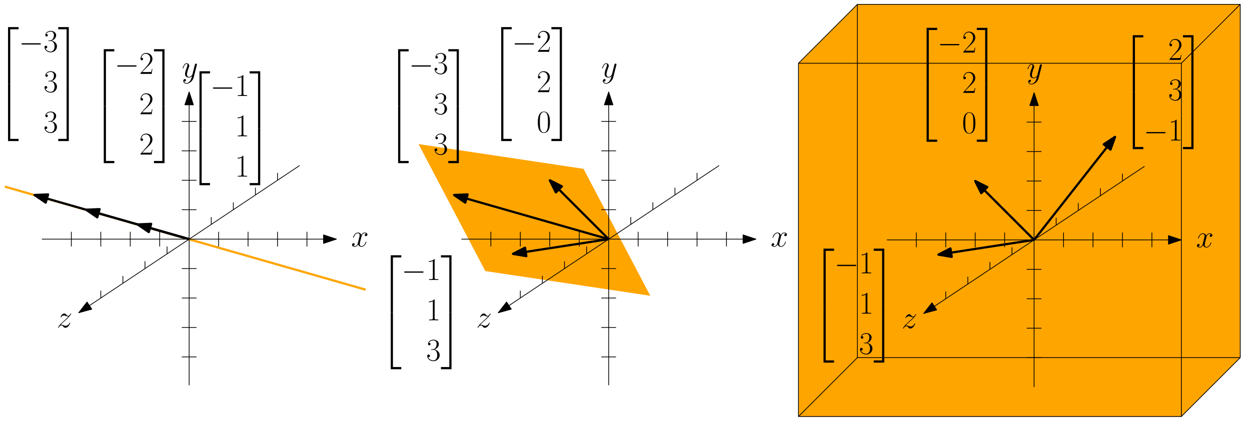

$\begin{bmatrix}-3\\3\\3\end{bmatrix}$
$\begin{bmatrix}-2\\2\\2\end{bmatrix}$
$\begin{bmatrix}-1\\1\\1\end{bmatrix}$
y
x
z
$\begin{bmatrix}-3\\3\\3\end{bmatrix}$
$\begin{bmatrix}-2\\2\\0\end{bmatrix}$
$\begin{bmatrix}-1\\1\\3\end{bmatrix}$
y
x
z
$\begin{bmatrix}-2\\2\\0\end{bmatrix}$
$\begin{bmatrix}2\\3\\-1\end{bmatrix}$
$\begin{bmatrix}-1\\1\\3\end{bmatrix}$
y
x
z

Figure 1.22: The span of three vectors in $\mathbb{R}^3$: a line (left), a plane (middle), the whole space (right).

We always have $\mathbf{0} \in \mathbf{Span}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$ (obtained by setting all λ_j to 0). This even holds if $n = 0$ and there are no vectors. As we have argued in Section 1.1.5, $\mathbf{0}$ is a linear combination of the empty sequence of vectors (and the only one), so $\mathbf{Span}() = \{\mathbf{0}\}$.

In "span language," Fact 1.5 can be rewritten as follows:

$$\mathbf{Span}\left(\begin{bmatrix}2\\3\end{bmatrix}, \begin{bmatrix}3\\-1\end{bmatrix}\right) = \mathbb{R}^2.$$

The span of two nonzero vectors that are scalar multiples of each other is always a line. For example,

$$\mathbf{Span}\left(\begin{bmatrix}2\\3\end{bmatrix}, \begin{bmatrix}4\\6\end{bmatrix}\right) = \left\{\lambda \begin{bmatrix}2\\3\end{bmatrix} : \lambda \in \mathbb{R}\right\}.$$

Figure 1.23 illustrates these two cases.

Next we prove a useful statement that may seem obvious from the examples in $\mathbb{R}^3$ but needs a proof: the span of vectors does not change when we add a linear combination of them as a new vector.

Lemma 1.23. *Let* $\mathbf{v}_1, \mathbf{v}_2 \ldots, \mathbf{v}_n \in \mathbb{R}^m$*, and let* $\mathbf{v} \in \mathbb{R}^m$ *be a linear combination of* $\mathbf{v}_1, \mathbf{v}_2 \ldots, \mathbf{v}_n$*. Then*

$$\underbrace{\mathbf{Span}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)}_{S} = \underbrace{\mathbf{Span}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n, \mathbf{v})}_{T}.$$

To prove that two sets S and T are equal, we can argue that each element of S is contained in T (then S is a *subset* of T, in formulas $S \subseteq T$) and—vice versa—that each element of T is contained in S (then $T \subseteq S$). Having done this, there can't be an element which is in one of the sets and not in the other one (same logic as with the implications in Lemma 1.19), so the two sets must be equal.

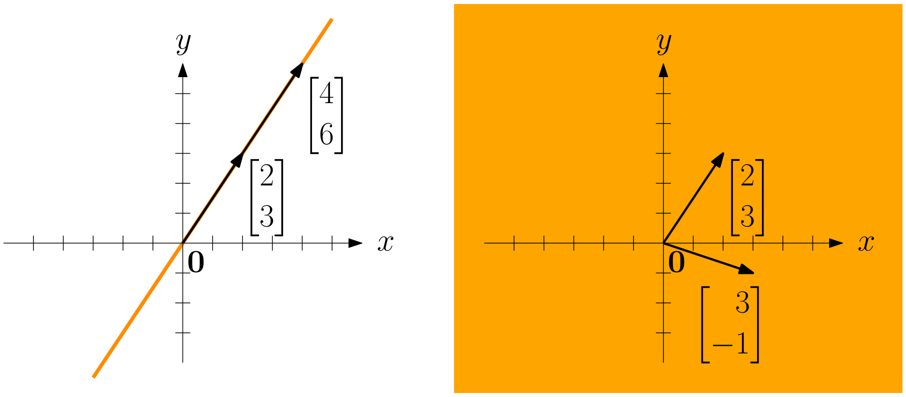

y
x
$\mathbf{0}$
$\begin{bmatrix}4\\6\end{bmatrix}$
$\begin{bmatrix}2\\3\end{bmatrix}$
y
x
$\mathbf{0}$
$\begin{bmatrix}2\\3\end{bmatrix}$
$\begin{bmatrix}3\\-1\end{bmatrix}$

Figure 1.23: The span of two vectors in $\mathbb{R}^2$: a line (left), or the whole space (right).

Proof.

$S \subseteq T$: Each element $\mathbf{w} \in S$ is a linear combination of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ and therefore also a linear combination of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n, \mathbf{v}$ (add the scalar multiple $0\mathbf{v}$). So $\mathbf{w} \in T$.

$T \subseteq S$: each element $\mathbf{w} \in T$ is a linear combination of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n, \mathbf{v}$,

$$\mathbf{w} = \sum_{j=1}^{n} \lambda_j \mathbf{v}_j + \lambda \mathbf{v}.$$

But since $\mathbf{v}$ is a linear combination of $\mathbf{v}_1, \mathbf{v}_2 \ldots, \mathbf{v}_n$, we also have

$$\mathbf{v} = \sum_{j=1}^{n} \mu_j \mathbf{v}_j.$$

Plugging the second equation into the first one, we get

$$\mathbf{w} = \sum_{j=1}^{n} \lambda_j \mathbf{v}_j + \lambda \mathbf{v} = \sum_{j=1}^{n} \lambda_j \mathbf{v}_j + \lambda \left(\sum_{j=1}^{n} \mu_j \mathbf{v}_j \right) = \sum_{j=1}^{n} (\lambda_j + \lambda \mu_j) \mathbf{v}_j.$$

This means that $\mathbf{w}$ is a linear combination of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ and hence in S. □

Chapter 2

Matrices

2.1 Matrices and linear combinations

A matrix is a rectangular array of numbers. Upfront, this is just a notation for a sequence of column vectors (the columns of the matrix), or a sequence of row vectors (the rows of the matrix). But matrices turn out to be very useful in representing, arguing about or computing with sequences of vectors and their linear combinations. Central matrix concepts that we introduce here are matrix-vector multiplication, the column space, the row space, the transpose, and the rank.

We often work with sequences of vectors. A *matrix* can be considered as a more compact notation for such a sequence. For example,

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \text{ is a } 3 \times 2 \text{ matrix (3 rows, 2 columns) that represents the sequence } \left(\begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}\right)$$

of 2 vectors in $\mathbb{R}^3$. Using the matrix, we save brackets and commas, but more importantly, it naturally represents a *second* sequence of 3 (row) vectors in $\mathbb{R}^2$:

$$\left(\begin{bmatrix} 1 & 2 \end{bmatrix}, \begin{bmatrix} 3 & 4 \end{bmatrix}, \begin{bmatrix} 5 & 6 \end{bmatrix}\right).$$

Definition 2.1 (Matrix). *An $m \times n$ matrix is a rectangular array of real numbers with m rows and n columns. We use upper-case letters ($A, B, \ldots$) to denote matrices, and write their entries with the corresponding lower case letters and two indices, as in*

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

Hence, a_{ij} is the entry in row i and column j of matrix A. The "dot-free" notation (see also Section 1.1.5) is

$$A = [a_{ij}]_{i=1,j=1}^{m\ \ n}.$$

The set of $m \times n$ matrices is denoted by $\mathbb{R}^{m \times n}$.

If we want to talk about the columns of A as column vectors or the rows of A as row vectors, we use column notation *or* row notation,

$$\underbrace{A = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & & | \end{bmatrix}}_{\text{column notation}}, \qquad \underbrace{A = \begin{bmatrix} — & \mathbf{u}_1 & — \\ — & \mathbf{u}_2 & — \\ & \vdots & \\ — & \mathbf{u}_m & — \end{bmatrix}}_{\text{row notation}}.$$

While a vector needs one dot symbol in "dot notation", a matrix needs seven. This is significant enough to use the other notations (dot-free, column, row) more frequently for matrices.

A column vector $\mathbf{v} \in \mathbb{R}^m$ is an $m \times 1$ matrix, and a row vector $\mathbf{u} \in \mathbb{R}^n$ is a $1 \times n$ matrix.

At this point, we have to admit an abuse of notation that we have started much earlier. While $\mathbb{R}^m$ officially contains sequences $(x_1, x_2 \ldots, x_m)$ of real numbers, we have silently also treated column vectors with m entries as elements of $\mathbb{R}^m$. In reality, they are matrices: elements of $\mathbb{R}^{m \times 1}$. But since an $m \times 1$ matrix just contains a sequence of m real numbers, there is no harm in treating it as an element of $\mathbb{R}^m$ as well. The same is true for row vectors: there is not really a difference between $\mathbb{R}^{1 \times n}$ and $\mathbb{R}^n$. While we are at it, there is also no problem in treating a 1×1 matrix as a real number.

Just like vectors, two matrices of the same shape can be added and multiplied with a scalar, as in the following examples:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}, \quad 2\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}.$$

Definition 2.2 (Matrix addition, scalar multiplication, zero matrix, square matrix)**.** *Let $A = [a_{ij}]_{i=1,j=1}^{m\ \ n}$ and $B = [b_{ij}]_{i=1,j=1}^{m\ \ n}$ be $m \times n$ matrices, $\lambda \in \mathbb{R}$ a scalar.*

(i) The matrix $A + B := [a_{ij} + b_{ij}]_{i=1,j=1}^{m\ \ n}$ is the sum *of A and B.*

(ii) The matrix $\lambda A := [\lambda a_{ij}]_{i=1,j=1}^{m\ \ n}$ is a scalar multiple *of A.*

(iii) The matrix $[0]_{i=1,j=1}^{m\ \ n}$ is the $m \times n$ zero matrix, *written as 0.*

(iv) If $m = n$ (number of rows equals number of columns), then A is a square matrix.

The non-square matrices come in two kinds of shapes, with somewhat unofficial but intuitive names. We have the *tall and skinny* matrices with more rows than columns, and the *short and wide* matrices with more columns than rows; see Figure 2.1.

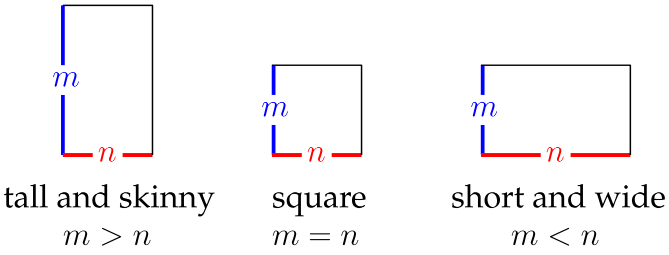
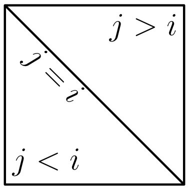


Figure 2.1: Matrix shapes

Square matrices are particularly important, and they often have additional properties. Before we provide the general definitions, we give some 3×3 examples.

$$\begin{array}{ccccc}
\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} &
\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 5 \end{bmatrix} &
\begin{bmatrix} 2 & 1 & 0 \\ 0 & 4 & 7 \\ 0 & 0 & 5 \end{bmatrix} &
\begin{bmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 0 & 7 & 5 \end{bmatrix} &
\begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 7 \\ 0 & 7 & 5 \end{bmatrix} \\
& & & & \\
\text{identity} & \text{diagonal} & \text{upper triangular} & \text{lower triangular} & \text{symmetric} \\
\text{matrix} & \text{matrix} & \text{matrix} & \text{matrix} & \text{matrix}
\end{array}$$

Definition 2.3 (Square matrix classes)**.** *Let $A = [a_{ij}]_{i=1,j=1}^{m\ \ m}$ be an $m \times m$ square matrix. If $j < i$, $j = i$, $j > i$, then a_{ij} is said to be* below, on, above *the diagonal.*



(i) *If $a_{ii} = 1$ for all i and $a_{ij} = 0$ for all $j \neq i$, then A is the* identity *matrix, denoted (in abuse of notation) by I in every dimension. A different way of defining I is as*

$$I := [\delta_{ij}]_{i=1,j=1}^{m\ \ m}.$$

Here, δ_{ij} is the Kronecker delta, *defined as 1 if $i = j$ and 0 otherwise.*

(ii) *If $a_{ij} = 0$ for all $j \neq i$ (entries not on the diagonal are 0), then A is a* diagonal *matrix.*

(iii) *If $a_{ij} = 0$ for all $j < i$ (entries below the diagonal are 0), then A is an* upper triangular *matrix.*

(iv) *If $a_{ij} = 0$ for all $j > i$ (entries above the diagonal are 0), then A is a* lower triangular *matrix.*

(v) *If $a_{ij} = a_{ji}$ for all i, j, then A is a* symmetric *matrix.*

Note that (ii)-(iv) each require that some entries are zero, but *not* that the other entries are nonzero. For example, the zero matrix is at the same time diagonal, upper triangular, and lower triangular, even though we do not see a "diagonal", or a "triangle" in it. Also whenever we say things like "all i", we mean all *applicable* i, in this case $i \in [m]$.

2.1.1 Matrix-vector multiplication

Here is the efficient "matrix way" of writing down a linear combination as a *matrix-vector multiplication*:

$$\underbrace{7\begin{bmatrix}1\\3\\5\end{bmatrix}+8\begin{bmatrix}2\\4\\6\end{bmatrix}}_{\text{linear combination}} \quad = \quad \underbrace{\begin{bmatrix}1&2\\3&4\\5&6\end{bmatrix}\begin{bmatrix}7\\8\end{bmatrix}}_{\text{matrix-vector product}}.$$

Definition 2.4 (Matrix-vector multiplication)**.** *Let*

$$A = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{m\times n}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n.$$

The vector

$$A\mathbf{x} := \sum_{j=1}^{n} x_j \mathbf{v}_j \in \mathbb{R}^m$$

is the product *of A and* $\mathbf{x}$*.*

It is important to also understand the product in the other matrix notations.

Observation 2.5. *Let*

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = [a_{ij}]_{i=1,j=1}^{m\ \ n} \in \mathbb{R}^{m\times n}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = [x_j]_{j=1}^{n} \in \mathbb{R}^n.$$

Then

$$A\mathbf{x} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix} = \left[\sum_{j=1}^{n} a_{ij}x_j\right]_{i=1}^{m} \in \mathbb{R}^m.$$

This is in many sources the official definition of matrix-vector multiplication. It does not explicitly refer to the columns or rows of A but just looks at A as a two-dimensional array of numbers. This is a very useful definition if we actually want to compute the product, but it hides our motivation for defining the product as a short notation for a linear combination.

But we easily see that both definitions say the same, by annotating the columns in Observation 2.5:

$$A = \begin{array}{c}\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \\ \begin{matrix} \mathbf{v}_1 & \quad\mathbf{v}_2 & \quad\cdots & \quad\mathbf{v}_n \end{matrix}\end{array}, \quad A\mathbf{x} = \begin{array}{c}\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix} \\ x_1\mathbf{v}_1 \; + x_2\mathbf{v}_2 \; + \cdots + x_n\mathbf{v}_n \end{array}.$$

An immediate consequence is the following.

Corollary 2.6. *Let I be the $m \times m$ identity matrix (Definition 2.3). Then $I\mathbf{x} = \mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^m$.*

Finally, we ce can also use row notation of A to define $A\mathbf{x}$ in terms of scalar products.

Observation 2.7. *Let*

$$A = \begin{bmatrix} — & \mathbf{u}_1 & — \\ — & \mathbf{u}_2 & — \\ & \vdots & \\ — & \mathbf{u}_m & — \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad \mathbf{x} \in \mathbb{R}^n. \quad \textit{Then } A\mathbf{x} = \underbrace{\begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{x} \\ \mathbf{u}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{u}_m \cdot \mathbf{x} \end{bmatrix}}_{\text{scalar products}}.$$

This is seen to be correct by annotating the *rows* in Observation 2.5:

$$A = \begin{array}{cc}\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} & \begin{matrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_m \end{matrix}\end{array}, \quad A\mathbf{x} = \begin{array}{cc}\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix} & \begin{matrix} \mathbf{u}_1 \cdot \mathbf{x} \\ \mathbf{u}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{u}_m \cdot \mathbf{x} \end{matrix}\end{array}.$$

Figure 2.2 provides another pictorial view, illustrating that if A is an $m \times n$ matrix, we can multiply it with an n-dimensional vector $\mathbf{x}$ and obtain an m-dimensional vector $A\mathbf{x}$ as the result.

2.1.2 Column space and rank

Definition 2.8 (Column space)**.** *Let A be an $m \times n$ matrix. The* column space $\mathbf{C}(A)$ *of A is the span (set of all linear combinations) of the columns,*

$$\mathbf{C}(A) := \{A\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\} \subseteq \mathbb{R}^m.$$

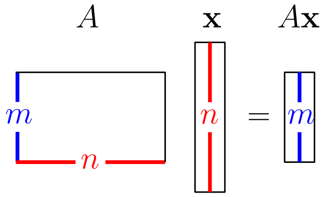


Figure 2.2: Matrix-vector multiplication, pictorially

Here we use the definition of the matrix-vector product $A\mathbf{x}$ as the linear combination of the columns with scalars from $\mathbf{x}$; see Definition 2.4. When we consider all possible vectors $\mathbf{x} \in \mathbb{R}^n$, we obtain all possible linear combinations (the span) of the columns. Note that we always have $\mathbf{0} \in \mathbf{C}(A)$.

Using the column space, the statement of Fact 1.5 can be written as

$$\mathbf{C}\left(\begin{bmatrix} 2 & 3 \\ 3 & -1 \end{bmatrix}\right) = \mathbf{Span}\left(\begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 3 \\ -1 \end{bmatrix}\right) = \mathbb{R}^2.$$

See Figure 1.23 (right) for an illustration. In general, when A is a 2×2 matrix with linearly independent columns, $\mathbf{C}(A) = \mathbb{R}^2$ holds; see Challenge 1.6.

A crucial parameter of a matrix is its rank. The rank is defined as the number of independent columns. A column is called independent if it is not a linear combination of previous columns.

Definition 2.9 ((In)dependent column and rank of a matrix)**.** *Let*

$$A = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & & | \end{bmatrix}$$

be an $m \times n$ matrix with columns $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$. Column $\mathbf{v}_j$ is called independent *if $\mathbf{v}_j$ is not a linear combination of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{j-1}$. Otherwise, $\mathbf{v}_j$ is called* dependent*. The* rank *of A, written as $\mathbf{rank}(A)$, is the number of independent columns of A.*

This means, $\mathbf{rank}(A)$ is a number between 0 and n. We have $\mathbf{rank}(A) = n$ exactly if no column is a linear combination of the previous ones. According to Corollary 1.20, this is the same as saying that the columns are linearly independent. The case $\mathbf{rank}(A) = 0$ happens exactly if $A = 0$, the zero matrix. Note that for $A = 0$, already the first column is a linear combination of the previous ones, because $\mathbf{0}$ is a linear combination of the empty sequence of vectors; see Section 1.1.5.

Two more examples (see Figure 2.3) are

$$\mathbf{rank}\left(\begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix}\right) = 2, \quad \mathbf{rank}\left(\begin{bmatrix} 2 & 4 \\ 3 & 6 \end{bmatrix}\right) = 1.$$

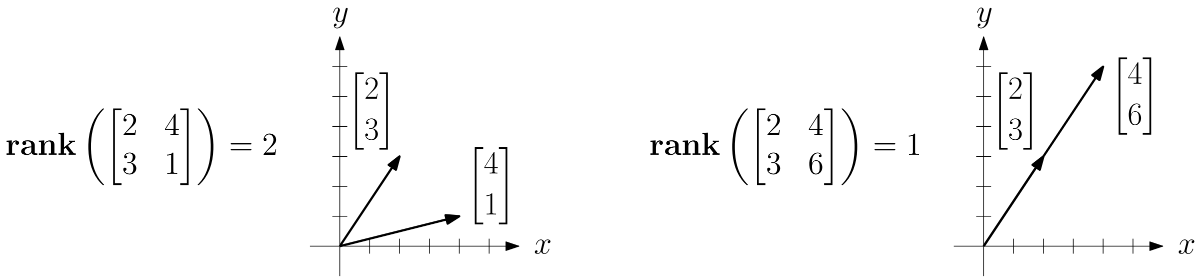

$\textbf{rank}\left(\begin{bmatrix}2 & 4\\3 & 1\end{bmatrix}\right) = 2$
y
$\begin{bmatrix}2\\3\end{bmatrix}$
$\begin{bmatrix}4\\1\end{bmatrix}$
x
$\textbf{rank}\left(\begin{bmatrix}2 & 4\\3 & 6\end{bmatrix}\right) = 1$
y
$\begin{bmatrix}2\\3\end{bmatrix}$
$\begin{bmatrix}4\\6\end{bmatrix}$
x

Figure 2.3: Ranks of two 2×2 matrices: 2 when both columns are independent (left), or 1 when only the first column is independent (right)

If we reorder the columns of a matrix, we may get other independent columns. For example, the two matrices

$$\begin{bmatrix}2 & 4\\3 & 6\end{bmatrix} \text{ and } \begin{bmatrix}4 & 2\\6 & 3\end{bmatrix}$$

have the same columns, but in different order. Each matrix has one independent column, namely its first one, and these are different. Still, both matrices have the same rank 1. This is not a coincidence. We will take this up again in the example immediately preceding Section 4.2.2; a consequence of the results in that section is that the rank of a matrix does not change when we reorder the columns.

The independent columns of a matrix A are also of interest, because they already span the column space of A.

Lemma 2.10. *Let A be an $m \times n$ matrix with r independent columns, and let C be the $m \times r$ submatrix containing the independent columns. Then $\mathbf{C}(A) = \mathbf{C}(C)$.*

By a *submatrix* of a matrix A, we mean a matrix obtained from A by deleting some rows and/or columns. Here, C is obtained from A by deleting the dependent columns.

Proof. Let $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r$ be the independent columns of A, and $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{n-r}$ the dependent columns (in the same order as they appear in A). We will prove that

$$\underbrace{\mathbf{Span}(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{n-r})}_{\mathbf{C}(A)} = \underbrace{\mathbf{Span}(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r)}_{\mathbf{C}(C)}.$$

We first observe that $\mathbf{w}_j$ is a linear combination of $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r, \mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_{j-1}$, for all j. Indeed, by Definition 2.9, a dependent column is a linear combination of the previous columns in A, and the sequence $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r, \mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_{j-1}$ contains all those (and possibly a few extra independent ones). Hence, if we start from the sequence $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r$ and then add $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{n-r}$ one by one, Lemma 1.23 guarantees that the span of the sequence never changes. □

$$\mathbf{R}\left(\begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix}\right) = \mathbb{R}^2$$

$$\mathbf{R}\left(\begin{bmatrix} 2 & 4 \\ 3 & 6 \end{bmatrix}\right) = \left\{c\begin{bmatrix} 2 & 4 \end{bmatrix} : c \in \mathbb{R}\right\}$$

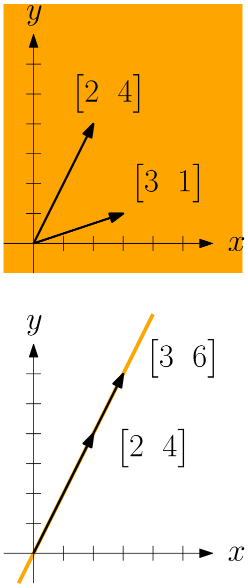


Figure 2.4: Row spaces of two 2×2 matrices: the whole plane (top) when the rows are linearly independent, or a line (botttom) when the rows are linearly dependent

2.1.3 Row space and transpose

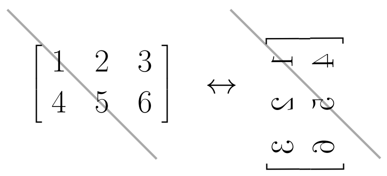
Recall that a matrix represents a second sequence of vectors, namely its rows. So we can also define the row space $\mathbf{R}(A)$ of a matrix: the span of its rows. Figure 2.4 illustrates the row spaces of the matrices from Figure 2.3.

For both matrices, the number of independent columns equals the number of independent rows. Is this a coincidence? No! It turns out that this is true for *every* matrix. This is quite surprising, and we will prove it in Section 4.3.2.

What we will do here is prepare the ground. We are still missing formal definitions of row space, independent rows, and potentially of a *row* rank based on counting independent rows. Conceptually, nothing new happens here, and we could easily provide "row versions" of Definitions 2.8 and 2.9 by essentially copying the "column versions". But mathematicians do not like this kind of copy & paste. The more elegant way is to consider the rows as the columns of another matrix, and thus reduce everything to the column versions.

This needs the concept of *matrix transposition*. The transpose $A^\top$ of a matrix A is another matrix, obtained by "mirroring" A along the *diagonal* "$\diagdown$", the line going through the diagonal entries $a_{11}, a_{22}, \ldots$ of A. Figure 2.5 shows a physical such mirror image and its mathematical abstraction where we don't screw up the number and bracket symbols. The effect of this mirroring is that the rows of A become the columns of $A^\top$.

Definition 2.11 (Transpose). *Let* $A = [a_{ij}]_{i=1,j=1}^{m\ \ n}$ *be an* $m \times n$ *matrix. The* transpose *of* A *is*



$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \leftrightarrow \quad A^\top = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Figure 2.5: Mirroring a matrix along the diagonal, physically and mathematically

the $n \times m$ matrix

$$A^\top := [a_{ji}]_{i=1,j=1}^{n \quad m}.$$

This means, the entry of $A^\top$ in row i and column j is a_{ji}, the entry of A in row j and column i. Transposing a matrix thus interchanges columns with rows. In particular, we can use transposition to turn column vectors into row vectors and vice versa, for example

$$\begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}^\top = \begin{bmatrix} 1 & 3 & 5 \end{bmatrix}.$$

In column and row notation, we therefore have

$$A = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & & | \end{bmatrix} \quad \Leftrightarrow \quad A^\top = \begin{bmatrix} \text{—} & \mathbf{v}_1^\top & \text{—} \\ \text{—} & \mathbf{v}_2^\top & \text{—} \\ & \vdots & \\ \text{—} & \mathbf{v}_n^\top & \text{—} \end{bmatrix},$$

$$A = \begin{bmatrix} \text{—} & \mathbf{u}_1 & \text{—} \\ \text{—} & \mathbf{u}_2 & \text{—} \\ & \vdots & \\ \text{—} & \mathbf{u}_m & \text{—} \end{bmatrix} \quad \Leftrightarrow \quad A^\top = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1^\top & \mathbf{u}_2^\top & \cdots & \mathbf{u}_m^\top \\ | & | & & | \end{bmatrix}.$$

It is easy to see that mirroring twice gives back the original. Also, the symmetric matrices are exactly the ones that are mirror images of themselves.

Observation 2.12. *Let A be an $m \times n$ matrix. Then*

$$(A^\top)^\top = A.$$

Moreover, a square matrix A is symmetric (Definition 2.3) if and only if $A = A^\top$.

Now we can define the row space of A simply as the column space of $A^\top$.

Definition 2.13 (Row space)**.** *Let A be an $m \times n$ matrix. The* row space $\mathbf{R}(A)$ *of A is the column space of the transpose,*

$$\mathbf{R}(A) := \mathbf{C}(A^\top).$$

Similarly, we could define an independent row of A as an independent column of $A^\top$, and the row rank of A as the (column) rank of $A^\top$, but there is no need to do this anymore. Using the transpose, we can formulate everything in terms of columns.

2.1.4 Rank-1 matrices

As a warmup, we will prove $\mathbf{rank}(A) = \mathbf{rank}(A^\top)$ for the case where A has rank 1. This will be a consequence of the following lemma that tells us how rank-1 matrices look like. Before that, let's look at an example of a rank-1-matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{bmatrix}.$$

This matrix has rank 1, since there is only one independent column (the first one), and the second and third are scalar multiples of it. In this example, there is also one independent row (the first one), and the second one is a scalar multiple of it.

Lemma 2.14. *Let A be an $m \times n$ matrix. The following two statements are equivalent.*

(i) $\mathbf{rank}(A) = 1$.

(ii) There are nonzero vectors $\mathbf{v} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n$ such that

$$A = [v_i w_j]_{i=1,j=1}^{m\ \ n}.$$

In dot notation, the rank-1 matrices are therefore exactly the nonzero matrices of the form

$$\begin{bmatrix} v_1w_1 & v_1w_2 & \cdots & v_1w_n \\ v_2w_1 & v_2w_2 & \cdots & v_2w_n \\ \vdots & \vdots & \ddots & \vdots \\ v_mw_1 & v_mw_2 & \cdots & v_mw_m \end{bmatrix}.$$

Proof. (i)⇒(ii): If $\mathbf{rank}(A) = 1$, there is exactly one independent column $\mathbf{v} \neq \mathbf{0}$ (Definition 2.9). This means, all columns before $\mathbf{v}$ are $\mathbf{0}$, and all columns after $\mathbf{v}$ are scalar multiples of $\mathbf{v}$. Thus, all columns are scalar multiples of $\mathbf{v}$ where at least one scalar (the one for column $\mathbf{v}$ itself) is nonzero. Let $\mathbf{w} \neq 0$ be the vector of scalars, meaning that the j-th column of A is $w_j\mathbf{v}$. Hence, a_{ij} (the entry in row i and column j of A) equals $w_jv_i = v_iw_j$. In other words,

$$A = [v_i w_j]_{i=1,j=1}^{m\ \ n}.$$

(ii)⇒(i): If $A = [v_i w_j]_{i=1,j=1}^{m\ \ n}$ for nonzero vectors $\mathbf{v}, \mathbf{w}$, then the j-th column of A is $w_j\mathbf{v}$. The first column for which $w_j \neq 0$ is independent, since $\mathbf{v} \neq \mathbf{0}$ and all columns before it are $\mathbf{0}$; all columns after it are scalar multiples of this independent column (the scalar for column k is w_k/w_j). Hence, there is exactly one independent column, so $\mathbf{rank}(A) = 1$. □

Corollary 2.15. *Let A be an $m \times n$ matrix with* $\mathbf{rank}(A) = 1$. *Then also* $\mathbf{rank}(A^\top) = 1$.

Proof. By Lemma 2.14, there are $\mathbf{v}, \mathbf{w} \neq \mathbf{0}$ such that

$$A = [\underbrace{v_i w_j}_{a_{ij}}]_{i=1,j=1}^{m\ \ n}.$$

By Definition 2.11 of the transpose,

$$A^\top = [\underbrace{v_j w_i}_{a_{ji}}]_{i=1,j=1}^{n \quad m} = [w_i v_j]_{i=1,j=1}^{n \quad m},$$

with $\mathbf{w}, \mathbf{v} \neq \mathbf{0}$. Using Lemma 2.14 "in the other direction," we have $\mathbf{rank}(A^\top) = 1$. □

2.2 Matrix multiplication

> Matrix-vector multiplication is only a special case of a more powerful operation, namely *matrix-matrix multiplication* that we get to know in this section. This has many important applications, including matrix decompositions where we write a matrix as a product of other matrices with the goal of revealing some structural properties. We present the CR decomposition in this section; others will appear in subsequent chapters.

The matrix-vector product $A\mathbf{x}$ is a short notation for the linear combination of the columns of A with scalars from $\mathbf{x}$ (Definition 2.4).

Often, we consider several linear combinations of the same vectors; as an example, see Table 1.1 that computes three different linear combinations of two fixed vectors. We would also like to express this in matrix notation.

So let's suppose we have some fixed vectors (the columns of A) and several vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_b$ of scalars with the resulting linear combinations $A\mathbf{x}_1, A\mathbf{x}_2, \ldots, A\mathbf{x}_b$. To compactly represent them, we introduce a matrix B with columns $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_b$, and define the *matrix-matrix product* AB as the matrix with columns $A\mathbf{x}_1, A\mathbf{x}_2, \ldots, A\mathbf{x}_b$.

Definition 2.16 (Matrix multiplication)**.** *Let A be an $a \times n$ matrix and*

$$B = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_b \\ | & | & & | \end{bmatrix}$$

an $n \times b$ matrix. The $a \times b$ matrix

$$AB := \begin{bmatrix} | & | & & | \\ A\mathbf{x}_1 & A\mathbf{x}_2 & \cdots & A\mathbf{x}_b \\ | & | & & | \end{bmatrix}$$

is the product *of A and B.*

This allows us to summarize the computations of Table 1.1 in a single *matrix multiplication*; see Table 2.1.

Some comments are in order here: for AB to be defined, the number of columns of A needs to match the number of rows of B; this is the number n in the definition. This

$$\mathbf{v} = \begin{bmatrix}2\\3\end{bmatrix}, \mathbf{w} = \begin{bmatrix}3\\-1\end{bmatrix} :$$

λ	μ	$\lambda\mathbf{v} + \mu\mathbf{w}$
-3	2	$\begin{bmatrix}0\\-11\end{bmatrix}$
1	-1	$\begin{bmatrix}-1\\4\end{bmatrix}$
3	0	$\begin{bmatrix}6\\9\end{bmatrix}$

$$\underbrace{\begin{bmatrix}2 & 3\\3 & -1\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}-3 & 1 & 3\\2 & -1 & 0\end{bmatrix}}_{B} = \underbrace{\begin{bmatrix}0 & -1 & 6\\-11 & 4 & 9\end{bmatrix}}_{AB}$$

Table 2.1: Three linear combinations of two vectors $\mathbf{v}, \mathbf{w}$ (left), summarized in one matrix multiplication (right)

comes from the fact that the matrix-vector products $A\mathbf{x}_j$ are only defined if $\mathbf{x}_j \in \mathbb{R}^n$; see Definition 2.4. This means, B needs to have n rows. By the same definition, $A\mathbf{x}_j \in \mathbb{R}^a$ for all j, and this also explains why AB is an $a \times b$ matrix.

One can define the product AB directly, without referring to matrix-vector multiplication: in order to compute the entry of AB in row i and column j, we take the scalar product of the i-th row of A and the j-th column of B.

Observation 2.17. *Let*

$$A = \begin{bmatrix} — & \mathbf{u}_1 & — \\ — & \mathbf{u}_2 & — \\ & \vdots & \\ — & \mathbf{u}_a & — \end{bmatrix} \in \mathbb{R}^{a\times n}, \quad B = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_b \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n\times b}.$$

Then

$$AB = \underbrace{\begin{bmatrix} \mathbf{u}_1\cdot\mathbf{x}_1 & \mathbf{u}_1\cdot\mathbf{x}_2 & \cdots & \mathbf{u}_1\cdot\mathbf{x}_b \\ \mathbf{u}_2\cdot\mathbf{x}_1 & \mathbf{u}_2\cdot\mathbf{x}_2 & \cdots & \mathbf{u}_2\cdot\mathbf{x}_b \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_a\cdot\mathbf{x}_1 & \mathbf{u}_a\cdot\mathbf{x}_2 & \cdots & \mathbf{u}_a\cdot\mathbf{x}_b \end{bmatrix}}_{ab \text{ scalar products}} = [\mathbf{u}_i\cdot\mathbf{x}_j]_{i=1,j=1}^{a\quad b} \in \mathbb{R}^{a\times b}.$$

This is because the j-th column of AB is $A\mathbf{x}_j = [\mathbf{u}_i\cdot\mathbf{x}_j]_{i=1}^a$; see Observation 2.7. Finally, here is the dot-free definition of AB.

Observation 2.18. *Let* $A = [a_{ij}]_{i=1,j=1}^{a\quad n}$, $B = [b_{ij}]_{i=1,j=1}^{n\quad b}$. *Then*

$$AB = \left[\sum_{\ell=1}^{n} a_{i\ell}b_{\ell j}\right]_{i=1,j=1}^{a\quad b}.$$

This is just a different way of writing Observation 2.17, since $\sum_{\ell=1}^n a_{i\ell}b_{\ell j}$ equals $\mathbf{u}_i\cdot\mathbf{x}_j$ ("i-th row of A times j-th column of B").

Here are two examples. Let

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

Then

$$\begin{aligned} AB &= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1\cdot 0 + 2\cdot 1 & 1\cdot 1 + 2\cdot 0 \\ 3\cdot 0 + 4\cdot 1 & 3\cdot 1 + 4\cdot 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix} \quad \text{("column exchange in } A\text{")} \\ BA &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 0\cdot 1 + 1\cdot 3 & 0\cdot 2 + 1\cdot 4 \\ 1\cdot 1 + 0\cdot 3 & 1\cdot 2 + 0\cdot 4 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix} \quad \text{("row exchange in } A\text{")} \end{aligned}$$

We see that matrix multiplication is not commutative, since we may have $BA \neq AB$. If A and B are not square matrices, it may happen that AB is defined and BA is not. Or that both products are defined, but are of different shape.

It's not hard to understand the situation exactly. For AB to be defined, A must be $a \times n$ and B must be $n \times b$, for some number n. For BA to be defined as well, we also need $a = b$. This means, A must be $m \times n$ and B must be $n \times m$, for some number m. Then AB is $m \times m$ and BA is $n \times n$, so both products are square matrices. If $m = n$, they have the same shape, otherwise, one is larger than the other; Figure 2.6 provides a pictorial view of matrix multiplication.

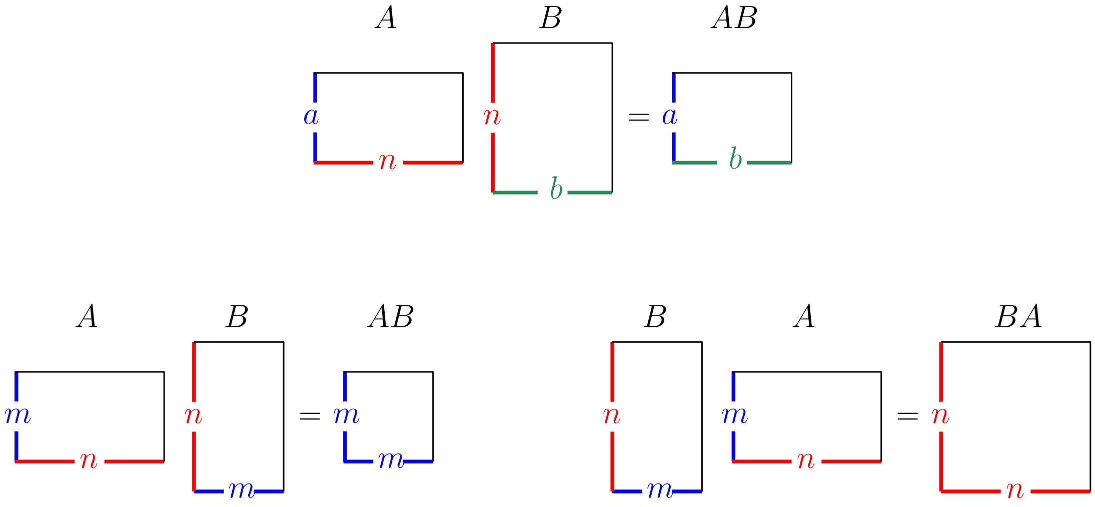


Figure 2.6: Matrix multiplication, pictorially

An easy but important fact is how matrix multiplication interacts with transposition (Definition 2.11).

Lemma 2.19. *Let A be an $a \times n$ matrix and B an $n \times b$ matrix. Then*

$$(AB)^\top = B^\top A^\top.$$

Proof. Since $B^\top$ is a $b \times n$ matrix and $A^\top$ an $n \times a$ matrix, the product $B^\top A^\top$ is a $b \times a$ matrix and therefore has the same shape as $(AB)^\top$.

Next, we compare the two matrices

$$(AB)^\top = [c_{ij}]_{i=1,j=1}^{b \quad a} \text{ and } B^\top A^\top = [d_{ij}]_{i=1,j=1}^{b \quad a}$$

entry by entry. By Definition 2.11 of the transpose, c_{ij} is the entry in row j and column i of AB and therefore the scalar product of the j-th row of A and the i-th column of B; see Observation 2.17:

$$c_{ij} = (j\text{-th row of } A) \cdot (i\text{-th column of } B).$$

By the same observation,

$$d_{ij} = (i\text{-th row of } B^\top) \cdot (j\text{-th column of } A^\top).$$

To conclude $c_{ij} = d_{ij}$, it remains to note that

$$(j\text{-th row of } A) = (j\text{-th column of } A^\top) \text{ and } (i\text{-th column of } B) = (i\text{-th row of } B^\top).$$

□

As seen before in Corollary 2.6 for matrix-vector multiplication, identity matrices also have no effect in matrix-matrix multiplications. We leave the proof of this as an (easy) exercise.

Corollary 2.20. *Let I be the $m \times m$ identity matrix. Then $IA = A$ for all $m \times n$ matrices, and $AI = A$ for all $n \times m$ matrices.*

2.2.1 Everything is matrix multiplication

You may already have realized that matrix-vector multiplication is a special case of matrix multiplication when we consider the vector as an $m \times 1$ matrix:

$$\underbrace{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}}_{2\times 2} \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{2\times 1} = \underbrace{\begin{bmatrix} 3 \\ 7 \end{bmatrix}}_{2\times 1}.$$

Similarly, we can now also talk about *vector-matrix multiplication* involving a row vector:

$$\underbrace{\begin{bmatrix} 1 & 1 \end{bmatrix}}_{1\times 2} \underbrace{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}}_{2\times 2} = \underbrace{\begin{bmatrix} 4 & 6 \end{bmatrix}}_{1\times 2}.$$

The scalar product of two vectors is another special case when we use the convention that 1×1 matrices can be treated as numbers. Writing the first vector as a row vector and the second one as a column vector, the scalar product becomes a matrix-matrix product:

$$\underbrace{\begin{bmatrix} 1 & 2 \end{bmatrix}}_{1\times 2} \underbrace{\begin{bmatrix} 3 \\ 4 \end{bmatrix}}_{2\times 1} = \underbrace{\begin{bmatrix} 11 \end{bmatrix}}_{1\times 1} = 11.$$

This is the reason why the scalar product $\mathbf{v} \cdot \mathbf{w}$ is often written as $\mathbf{v}^\top \mathbf{w}$. Formally, this is a matrix multiplication where we treat the resulting 1×1 matrix as a number.

There is another interesting variant here, the *outer product* that we get by multiplying a *column* vector with a *row* vector (they don't have to be of the same dimensions).

$$\underbrace{\begin{bmatrix} 3 \\ 4 \end{bmatrix}}_{2\times 1} \underbrace{\begin{bmatrix} 1 & 2 \end{bmatrix}}_{1\times 2} = \underbrace{\begin{bmatrix} 3 & 6 \\ 4 & 8 \end{bmatrix}}_{2\times 2}.$$

This provides another way of thinking about rank-1 matrices.

Lemma 2.21. *Let A be an $m \times n$ matrix. The following two statements are equivalent.*

(i) $\mathbf{rank}(A) = 1$.

(ii) There are nonzero vectors $\mathbf{v} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n$ such that A is their outer product, $A = \mathbf{v}\mathbf{w}^\top$.

Proof. By definition of matrix multiplication, $A = \mathbf{v}\mathbf{w}^\top$ means that entry a_{ij} of A is "the i-th row of $\mathbf{v}$ times the j-th column of $\mathbf{w}^\top$." The i-th row of $\mathbf{v}$ consists of just one number v_i, and similarly, the j-th column of $\mathbf{w}^\top$ has one number w_j. Hence, $A = \mathbf{v}\mathbf{w}^\top$ is equivalent to $a_{ij} = v_i w_j$ for all i and j, and this in turn is equivalent to the condition for rank 1 in Lemma 2.14. □

Definition 2.16 has introduced matrix multiplication via column notation and matrix-vector products. Using vector-matrix products, we can also write AB in row notation:

$$\underbrace{\begin{bmatrix} — & \mathbf{u}_1 B & — \\ — & \mathbf{u}_2 B & — \\ & \vdots & \\ — & \mathbf{u}_m B & — \end{bmatrix}}_{AB\text{, row notation}} = \underbrace{\begin{bmatrix} — & \mathbf{u}_1 & — \\ — & \mathbf{u}_2 & — \\ & \vdots & \\ — & \mathbf{u}_m & — \end{bmatrix}}_{A\text{, row notation}} \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \\ | & | & & | \end{bmatrix}}_{B\text{, column notation}} = \underbrace{\begin{bmatrix} | & | & & | \\ A\mathbf{x}_1 & A\mathbf{x}_2 & \cdots & A\mathbf{x}_n \\ | & | & & | \end{bmatrix}}_{AB\text{, column notation}}$$

2.2.2 Distributivity and associativity

While matrix multiplication is not commutative (we may have $AB \neq BA$), two other important laws hold.

Lemma 2.22. *Let A, B, C be three matrices such that all sums and products in the following are defined. Then*

(i) $A(B + C) = AB + AC$ *and* $(B + C)D = BD + CD$ (distributivity);

(ii) $(AB)C = A(BC)$ (associativity).

Proof. Unfortunately, this requires some work but is at the same time boring. We are therefore lazy and omit the proof of distributivity (it's also not that you couldn't find this anywhere else). The proof of associativity is a bit more interesting, as it lets us exercise (and appreciate) the dot-free definition of matrix multiplication; see Observation 2.18.

We define $S = (AB)C$ and compute s_{ij}, the entry of S in row i and column j. Then we do the same with $T = A(BC)$ and check that $s_{ij} = t_{ij}$ for all i and j.

In order for AB and BC to be defined, we need $A \in \mathbb{R}^{a,m}, B \in \mathbb{R}^{m,n}, C \in R^{n,c}$, for some integers a, m, n, c. Then $AB \in \mathbb{R}^{a,n}, BC \in \mathbb{R}^{m,c}$ and $(AB)C, A(BC) \in \mathbb{R}^{a,c}$. So both $A(BC)$ and $A(BC)$ have the same shape.

We know that s_{ij} is the i-th row of $Q = AB$ times the j-th column of C, see Observation 2.18:

$$s_{ij} = \sum_{\ell=1}^{n} q_{i\ell} c_{\ell j}, \text{ where } q_{i\ell} = \sum_{k=1}^{m} a_{ik} b_{k\ell} \quad (i\text{-th row of } A \text{ times } \ell\text{-th column of } B)$$

by the same logic. Putting this together, we get

$$s_{ij} = \sum_{\ell=1}^{n} \left(\sum_{k=1}^{m} a_{ik} b_{k\ell} \right) c_{\ell j} = \sum_{\ell=1}^{n} \sum_{k=1}^{m} a_{ik} b_{k\ell} c_{\ell j},$$

using distributivity of the real numbers ("constant factors can be pulled into and out of sums").

For t_{ij}, the entry of $A(BC)$ in row i and column j, we argue in the same way. With $R = BC$, we have

$$t_{ij} = \sum_{k=1}^{m} a_{ik} r_{kj}, \text{ where } r_{kj} = \sum_{\ell=1}^{n} b_{k\ell} c_{\ell j} \quad (k\text{-th row of } B \text{ times } j\text{-th column of } C).$$

Hence,

$$t_{ij} = \sum_{k=1}^{m} a_{ik} \left(\sum_{\ell=1}^{n} b_{k\ell} c_{\ell j} \right) = \sum_{k=1}^{m} \sum_{\ell=1}^{n} a_{ik} b_{k\ell} c_{\ell j}.$$

Now we see that s_{ij} and t_{ij} only differ in the order of summation, and as addition over the reals is commutative, we have $s_{ij} = t_{ij}$. □

We silently used an important technique here, exchange of summation order:

$$\sum_{k=1}^{m} \sum_{\ell=1}^{n} \cdots = \sum_{\ell=1}^{n} \sum_{k=1}^{m} \cdots$$

If you haven't seen this before, it's useful to look at this in more detail. Both double sums go through all pairs (k, ℓ) where k goes through the range from 1 to m and ℓ goes through the range from 1 to n. The only difference is the order. The left double sum goes through the pairs in the order

$$(1,1), (1,2), \ldots, (1,n), (2,1), (2,2), \ldots, (2,n), \ldots.$$

This is because the inner sum goes through all ℓ for $k = 1$, then for $k = 2$ and so on. In contrast, the right double sum goes through the pairs in the order

$$(1,1),(2,1),\dots,(m,1),(1,2),(2,2),\dots,(m,2),\dots.$$

As a computer science student, you can also think of this in terms of nested loops. Depending on whether you increase k or ℓ in the outer loop, you will print the pairs (k,ℓ) in different orders:

```
// (1,1),(1,2),...,(1,n),(2,1),...
for (int k = 1; k <= m; k++) {
    for (int l = 1; l <= n; l++) {
        // print (k,l)
    }
}
```

```
// (1,1),(2,1),...,(m,1),(1,2),...
for (int l = 1; l <= n; l++) {
    for (int k = 1; k <= m; k++) {
        // print (k,l)
    }
}
```

Figure 2.7 illustrates this for $m = 4, n = 3$.

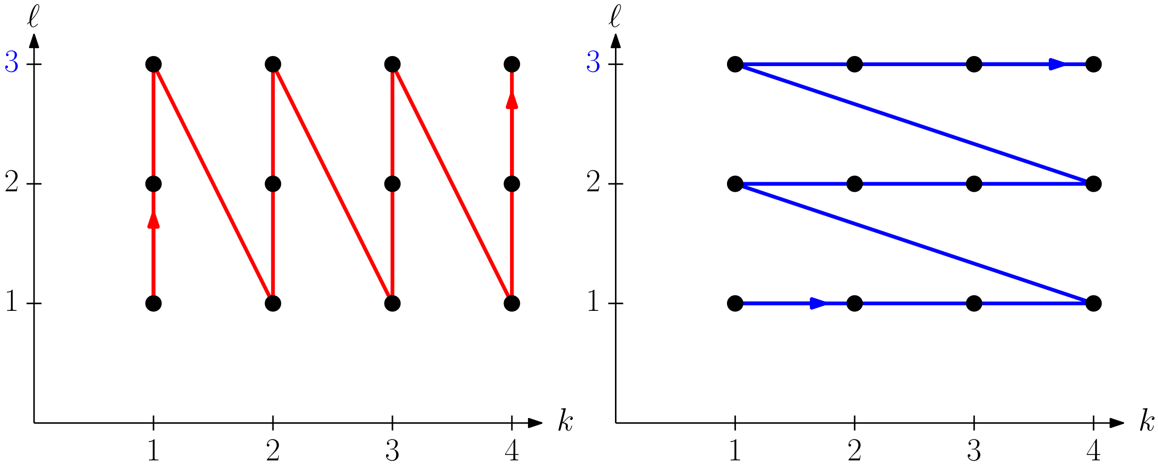


Figure 2.7: Different summation orders, same pairs: Outer sum / loop over k (left); outer sum / loop over ℓ (right)

Knowing that matrix multiplication is associative allows us to write ABC for a product of three matrices. As it doesn't matter whether we compute this as $(AB)C$ or $A(BC)$, we can as well omit the brackets.

This also works for more matrices and is then called *generalized associativity*. For example, $(AB)(CD) = A((BC)D) = \cdots = ABCD$, so we can again omit the brackets, since it doesn't matter where we put them. We will not prove this here (but informally in Section 2.3.4) and only point out that it is *not* obvious. Associativity in Lemma 2.22 only works for three matrices, and one needs a separate proof for more matrices. There are several such proofs [War01], but generalized associativity *does* need a proof.

2.2.3 CR decomposition

You know the prime factor decomposition of a natural number. For example, this writes the number 1001 as

$$1001 = 7 \cdot 11 \cdot 13.$$

This decomposition tells you something about the "structure" of the number that you cannot easily tell from just staring at the number.

In linear algebra, it's mostly matrix decompositions that are of interest. This means, we want to write a matrix A as a product of other matrices, with the goal of learning more about A, and potentially speeding up computations involving A.

In this section, we will see a first such decomposition.

Theorem 2.23 (CR decomposition)**.** *Let A be an $m \times n$ matrix of rank r (Definition 2.9). Let C be the $m \times r$ submatrix of A containing the independent columns. Then there exists a unique $r \times n$ matrix R such that*

$$A = CR.$$

Before we go to the proof, let us do an example. Consider the rank-1 matrix

$$A = \begin{bmatrix} 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}$$

with one independent column (the first one). Then the CR decomposition assumes the form of an outer product:

$$A = \underbrace{\begin{bmatrix} 2 \\ 3 \end{bmatrix}}_{C, 2\times 1} \underbrace{\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}}_{R, 1\times 3}.$$

The columns of R contain the scalars that we need in order to write each column as a scalar multiple of the first one. In fact, we have already proved that each rank-1 matrix can be written as an outer product, see Lemma 2.21. Next we show the existence of the CR decomposition for every matrix A.

Proof. We know that A and C have the same column space (Lemma 2.10), so every column of A can be written as a linear combination of the columns of C. Moreover, since the columns of C are linearly independent, the scalars in these linear combinations are unique (Lemma 1.21). If A has columns $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \in \mathbb{R}^m$, we therefore have $\mathbf{v}_j = C\mathbf{x}_j$ for all j, where $\mathbf{x}_j \in \mathbb{R}^r$ is a unique vector of r scalars. This uses that matrix-vector multiplication is simply another notation for linear combinations; see Definition 2.4. But then we get

$$A = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & & | \end{bmatrix} = C \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \\ | & | & & | \end{bmatrix}}_{R \in \mathbb{R}^{r\times n}} = CR,$$

using Definition 2.16 of matrix multiplication. □

So we have a decomposition $A = CR$. The matrix C contains the independent columns of A and the matrix R contains the unique scalars that we need in order to write all columns as linear combinations of the independent columns.

Below is an example. Each of the four columns $\mathbf{v}_j, j = 1, 2, 3, 4$, is either independent (and then $\mathbf{v}_j = 1\mathbf{v}_j$ is the unique linear combination), or it is dependent and a unique linear combination of the previous independent columns.

$$\begin{array}{r|c|c|c|c}
 & \multicolumn{4}{c}{\text{columns of } A} \\
 & \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 \\
 & \| & \| & \| & \| \\
A = \begin{bmatrix}1 & 2 & 0 & 3\\ 2 & 4 & 1 & 4\\ 3 & 6 & 2 & 5\end{bmatrix} & \begin{bmatrix}1\\2\\3\end{bmatrix} & \begin{bmatrix}2\\4\\6\end{bmatrix} & \begin{bmatrix}0\\1\\2\end{bmatrix} & \begin{bmatrix}3\\4\\5\end{bmatrix} \\
 & \| & \| & \| & \| \\
\mathbf{v}_1 \qquad\qquad\qquad & 1\mathbf{v}_1 & 2\mathbf{v}_1 & & 3\mathbf{v}_1 \\
\mathbf{v}_2 \qquad\quad & & & & \\
\mathbf{v}_3 \quad\; & & & 1\mathbf{v}_3 & -2\mathbf{v}_3 \\
\mathbf{v}_4 & & & & \\
\text{independent?} & \text{yes} & \text{no} & \text{yes} & \text{no}
\end{array}$$

The resulting CR decomposition is

$$\underbrace{\begin{bmatrix}1 & 2 & 0 & 3\\ 2 & 4 & 1 & 4\\ 3 & 6 & 2 & 5\end{bmatrix}}_{A,\ 3\times 4} = \underbrace{\begin{bmatrix}1 & 0\\ 2 & 1\\ 3 & 2\end{bmatrix}}_{C,\ 3\times 2} \underbrace{\begin{bmatrix}1 & 2 & 0 & 3\\ 0 & 0 & 1 & -2\end{bmatrix}}_{R,\ 2\times 4}.$$

You can check that the decomposition is correct, but we have not explained how it was computed. In this example, only the last column of R requires some actual computations: we need to write a vector as a linear combination of two independent vectors in $\mathbb{R}^2$. As in the proof of Fact 1.5, this boils down to solving a system of two linear equations in two variables. Easy enough, but if the matrix A is larger, we would need to solve larger systems in order to compute the CR decomposition. We don't yet know how to do this. Also, it is not clear at this point what exactly we learn about A from the decomposition $A = CR$.

We will come back to this in Section 3.5.5 where we will rediscover (and also efficiently compute) the CR decomposition in a different context (Gauss-Jordan elimination). Section 4.3 will interpret C and R in yet another context (C: basis of column space, R: basis of row space).

Exercise 2.24. *What are the matrices C and R in Theorem 2.23 if A is an $m \times m$ matrix of rank m (the columns are linearly independent)? What are the matrices C and R if A is the $m \times n$ zero matrix? (The second question also requires you to think about $m \times 0$ and $0 \times n$ matrices. While these are not extremely relevant in practice, it is good to understand how our definitions and proofs also work for them.)*

2.3 Matrices and linear transformations

When we multiply an $m \times n$ matrix A with a vector $\mathbf{x}$ in $\mathbb{R}^n$, we get a "transformed" vector $A\mathbf{x} \in \mathbb{R}^m$. Here, we look at the properties of and the theory behind this (linear) transformation. Linear transformations are for example used to draw 3-dimensional objects in 2-dimensional space, and they have many other applications, some of which we will encounter in subsequent chapters. We will also see that matrix multiplication naturally appears when combining different linear transformations.

2.3.1 Matrices as functions

An $m \times n$ matrix A can be thought of as "transforming" an input vector $\mathbf{x} \in \mathbb{R}^n$ into an output vector $A\mathbf{x} \in \mathbb{R}^m$. Formally, this transformation is a function from $\mathbb{R}^n$ to $\mathbb{R}^m$. The "from" set is the *domain* of the function, and the "to" set is its *range*.

Definition 2.25 (Matrix as function). *Let A be an $m \times n$ matrix. $T_A : \mathbb{R}^n \to \mathbb{R}^m$ is the function defined by*

$$T_A(\underbrace{\mathbf{x}}_{\in\mathbb{R}^n}) = \underbrace{A\mathbf{x}}_{\in\mathbb{R}^m}.$$

For example, if $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, then

$$T_A\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix},$$

the function that swaps the coordinates of the input vector in $\mathbb{R}^2$.

Observation 2.26. *Let A be an $m \times n$ matrix, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$. Then*

(i) $T_A(\mathbf{x}+\mathbf{y}) = T_A(\mathbf{x}) + T_A(\mathbf{y})$ and

(ii) $T_A(\lambda\mathbf{x}) = \lambda T_A(\mathbf{x})$.

By combining (i) and (ii), we also get $T_A(\lambda\mathbf{x} + \mu\mathbf{y}) = \lambda T_A(\mathbf{x}) + \mu T_A(\mathbf{y})$.

Proof. After substituting the definition of T_A, (i) and (ii) simply say that $A(\mathbf{x}+\mathbf{y}) = A\mathbf{x}+A\mathbf{y}$ and $A(\lambda\mathbf{x}) = \lambda A\mathbf{x}$. Both equalities easily follow from the rules of vector addition (Definition 1.1), scalar multiplication (Definition 1.3) and matrix-vector multiplication (Observation 2.5). □

To understand what T_A does for a given matrix A, it is useful to not only look at individual inputs, but at a whole set of inputs. For a set of input vectors $X \subseteq \mathbb{R}^n$, we define $T_A(X) := \{T_A(\mathbf{x}) : \mathbf{x} \in X\}$ as the set of transformed output vectors. In the following examples, we see how different T_A's transform the standard unit vectors $\mathbf{e}_1, \mathbf{e}_2$, and a set $X \subseteq \mathbb{R}^2$ (gray L-shaped polygon); see Figure 2.8 (middle). In all examples, the vector

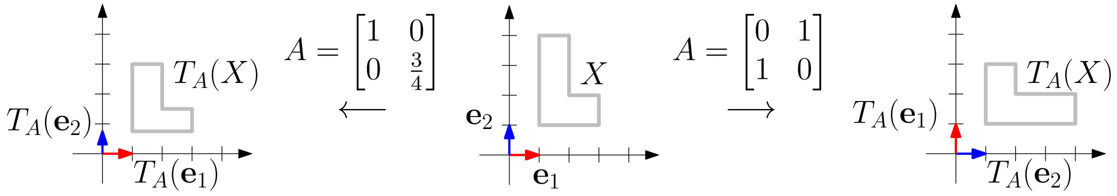


Figure 2.8: Right: Mirroring the input along the diagonal. Left: Stretching the input by a factor of $\frac{3}{4}$ along the second coordinate.

$T_A(\mathrm{e}_1)$ is the first column of A, while $T_A(\mathrm{e}_2)$ is the second column. The gray shape gets transformed accordingly.

The polygon X has 6 corners and 6 line segments connecting the corners. It is an infinite set, but to compute $T_A(X)$, we only need to apply T_A to the corners; the line segments will follow their two corners. To see this, consider a line segment s connecting two corners $\mathbf{x}$ and $\mathbf{y}$. In Section 1.1.4, we have seen that s is the set of convex combinations of $\mathbf{x}$ and $\mathbf{y}$,

$$s = \{\lambda\mathbf{x} + \mu\mathbf{y} : \lambda + \mu = 1, \lambda \geq 0, \mu \geq 0\}.$$

Hence,

$$\begin{array}{rcll} T_A(s) & = & \{T_A(\lambda\mathbf{x} + \mu\mathbf{y}) \quad : & \lambda + \mu = 1, \lambda \geq 0, \mu \geq 0\} \\ & = & \{\lambda T_A(\mathbf{x}) + \mu T_A(\mathbf{y}) \ : & \lambda + \mu = 1, \lambda \geq 0, \mu \geq 0\} \quad \text{(by Observation 2.26).} \end{array}$$

This means, the transformed line segment $T_A(s)$ is simply the line segment connecting the transformed corners $T_A(\mathbf{x})$ and $T_A(\mathbf{y})$.

For our first example above where T_A swaps the coordinates, the geometric effect of T_A is shown in Figure 2.8 (right): Swapping the two coordinates corresponds to mirroring the input along the diagonal "$\diagup$" of the coordinate system.

Figure 2.8 (left) gives an example of *stretching*, which means to make the input longer or shorter along some or all of the coordinates. If the *stretching factors* are the same for all coordinates, we have a *scaling*, resulting in a larger or smaller copy of the input.

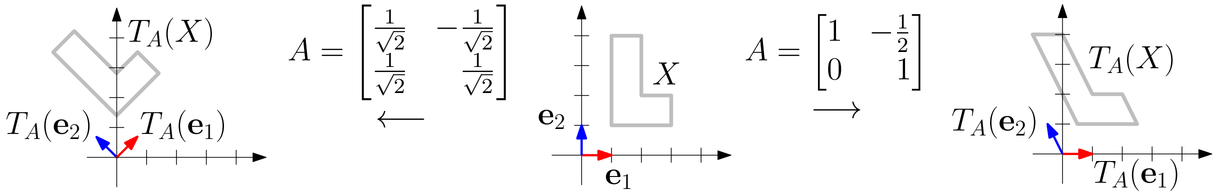
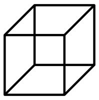

$T_A(X)$
$T_A(\mathbf{e}_2)$
$T_A(\mathbf{e}_1)$
$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$
$\longleftarrow$
X
$\mathbf{e}_2$
$\mathbf{e}_1$
$A = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & 1 \end{bmatrix}$
$\longrightarrow$
$T_A(X)$
$T_A(\mathbf{e}_2)$
$T_A(\mathbf{e}_1)$

Figure 2.9: Right: Shearing the input parallel to the first coordinate. Left: Rotating the input by 45 degrees.

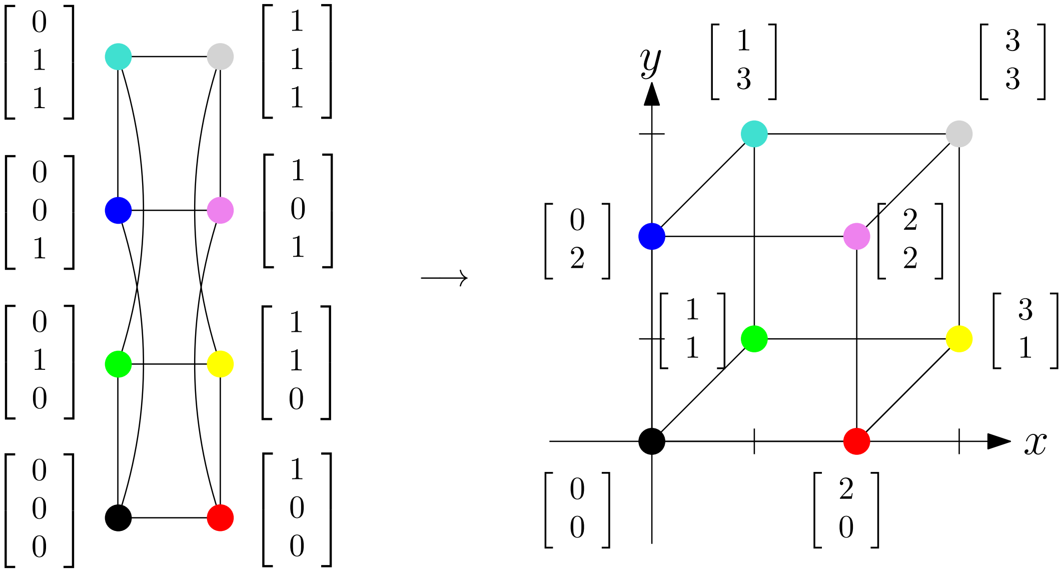
Figure 2.9 shows an example of a *shear*, and of a *rotation*. If $A = I$, the identity matrix, then T_A is the identity transformation, the function that simply outputs the input, without transforming it.

If A is a 2×3 matrix, then T_A is an *orthogonal projection* from $\mathbb{R}^3$ to $\mathbb{R}^2$. Such projections can be used to draw 3-dimensional objects in 2-dimensional space, for example the cube in the margin.



You can recognize the cube, although what you really see is just a 2-dimensional image. Figure 2.10 shows how such an image is obtained through a transformation T_A.

$$\text{Applying } T_A : \mathbb{R}^3 \to \mathbb{R}^2 \text{ with } A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 1 & 2 \end{bmatrix}:$$



$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$
$\longrightarrow$
y
$\begin{bmatrix} 1 \\ 3 \end{bmatrix}$
$\begin{bmatrix} 3 \\ 3 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 2 \end{bmatrix}$
$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 3 \\ 1 \end{bmatrix}$
x
$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$\begin{bmatrix} 2 \\ 0 \end{bmatrix}$

Figure 2.10: Orthogonal projection of a 3-dimensional cube. The left figure shows the 8 corners of the 3-dimensional *unit cube* as vectors in $\mathbb{R}^3$. Two corners are connected in the cube if they differ in exactly one coordinate. The right figure is a 2-dimensional drawing, resulting from applying T_A to the cube corners (the connections follow the corners). With different matrices A, we get different drawings; see Figure 2.11 for another drawing.

2.3.2 Linear transformations

After these examples, we are ready to define linear transformations in general.

Definition 2.27 (Linear transformation). *Let $T : \mathbb{R}^n \to \mathbb{R}^m$ be a function from $\mathbb{R}^n$ to $\mathbb{R}^m$. T is called a* linear transformation *if the following two statements hold for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and all $\lambda \in \mathbb{R}$.*

(i) $T(\mathbf{x} + \mathbf{y}) = T(\mathbf{x}) + T(\mathbf{y})$ *and*

(ii) $T(\lambda\mathbf{x}) = \lambda T(\mathbf{x})$.

By combining (i) and (ii), it then also holds that $T(\lambda\mathbf{x} + \mu\mathbf{y}) = \lambda T(\mathbf{x}) + \mu T(\mathbf{y})$.

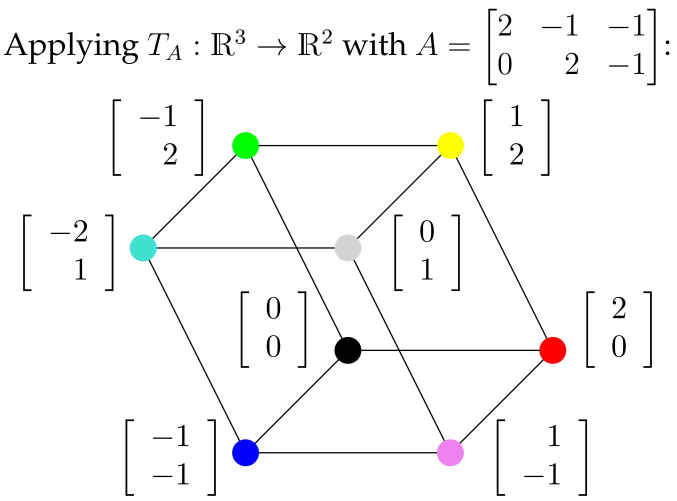

Applying $T_A : \mathbb{R}^3 \to \mathbb{R}^2$ with $A = \begin{bmatrix} 2 & -1 & -1 \\ 0 & 2 & -1 \end{bmatrix}$:
$\begin{bmatrix} -1 \\ 2 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$
$\begin{bmatrix} -2 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$\begin{bmatrix} 2 \\ 0 \end{bmatrix}$
$\begin{bmatrix} -1 \\ -1 \end{bmatrix}$
$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Figure 2.11: Projecting the cube in Figure 2.10 (left) using a different matrix

By Observation 2.26, all functions T_A in Definition 2.25 are linear transformations. The statements (i) and (ii) in Definition 2.27 are called the *axioms of a linear transformation*. An *axiom* is a statement that serves as a starting point for further reasoning and argument.[1]

What they say is the following: if we want to compute $T(\mathbf{x} + \mathbf{y})$, it doesn't matter whether we first add the two vectors and then apply T to the result, or whether we first apply T to both vectors and then add up the results. And in computing $T(\lambda\mathbf{x})$, we can first scale $\mathbf{x}$ by λ and then apply T to the result, but we get the same output when we first apply T to $\mathbf{x}$ and then scale the result by λ.

We can visualize this with *commutative diagrams*, see Table 2.2. In math jargon, these are also referred to as diagrams that commute. The arrows in a diagram correspond to certain operations, and if the diagram commutes, this means that we can follow any path through the diagram (apply the operations in any order), and the result is always the same.

Let us look at three examples and two counterexamples of linear transformations. The function $T : \mathbb{R}^n \to \mathbb{R}$ given by

$$T(\mathbf{x}) = \sum_{i=1}^{n} x_i$$

is a linear transformation. We can directly check that the axioms hold, or observe that T is of the form T_A for the $1 \times n$ matrix $A = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$ and then use Observation 2.26. Another (somewhat trivial but still instructive) example is $T : \mathbb{R}^n \to \mathbb{R}^m$ given by

$$T(\mathbf{x}) = \mathbf{0}.$$

Here, $\mathbf{0}$ is the m-dimensional zero vector; the axioms of linear transformations obviously hold, and as before, we could also write T in the form T_A for A the $m \times n$ zero matrix.

[1] `https://en.wikipedia.org/wiki/Axiom`, accessed September 7, 2024

$$\begin{tikzcd}
\mathbf{x}, \mathbf{y} \arrow[r, "T"] \arrow[d, "+"'] & T(\mathbf{x}), T(\mathbf{y}) \arrow[d, "+"] \\
\mathbf{x} + \mathbf{y} \arrow[r, "T"'] & T(\mathbf{x} + \mathbf{y}) = T(\mathbf{x}) + T(\mathbf{y})
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{x} \arrow[r, "T"] \arrow[d, "\cdot\lambda"'] & T(\mathbf{x}) \arrow[d, "\cdot\lambda"] \\
\lambda\mathbf{x} \arrow[r, "T"'] & T(\lambda\mathbf{x}) = \lambda T(\mathbf{x})
\end{tikzcd}$$

Table 2.2: Commutative diagrams for Definition 2.27 (linear transformation): In both cases, there are two ways to go from the top left to the bottom right, but the result is the same.

The *identity* $T : \mathbb{R}^n \to \mathbb{R}^n$ is given by $T(\mathbf{x}) = \mathbf{x}$. This is also linear and of the form T_A for $A = I$, the $n \times n$ identity matrix; see also Corollary 2.6.

Slightly changing the first example to

$$T(\mathbf{x}) = \sum_{i=1}^{n} |x_i| = \|\mathbf{x}\|_1$$

gives the 1-norm of $\mathbf{x}$ (see Section 1.2.2). This is no longer a linear transformation. To show this, we need to find a violation of the axioms. For example, if $\mathbf{x} \neq \mathbf{0}$ and $\lambda < 0$, then $T(\lambda\mathbf{x}) > 0$ but $\lambda T(\mathbf{x}) < 0$; so axiom (ii) does not always hold.

Slightly changing the second example, we can produce another counterexample. Consider

$$T(\mathbf{x}) = \mathbf{u},$$

where $\mathbf{u} \in \mathbb{R}^m$ is some fixed nonzero vector. Here, axiom (i), $T(\mathbf{x} + \mathbf{y}) = T(\mathbf{x}) + T(\mathbf{y})$, always fails, since $T(\mathbf{x} + \mathbf{y}) = \mathbf{u}$ and $T(\mathbf{x}) + T(\mathbf{y}) = 2\mathbf{u}$.

As an easy consequence of the axioms, every linear transformation T satisfies

$$T(\lambda\mathbf{x} + \mu\mathbf{y}) = \lambda T(\mathbf{x}) + \mu T(\mathbf{y}),$$

a point that we have already made in Definition 2.27. This generalizes to arbitrary linear combinations.

Lemma 2.28. *Let $T : \mathbb{R}^n \to \mathbb{R}^m$ be a linear transformation, let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\ell \in \mathbb{R}^n$ and $\lambda_1, \lambda_2, \ldots, \lambda_\ell \in \mathbb{R}$. Then*

$$T\left(\sum_{j=1}^{\ell} \lambda_j \mathbf{x}_j\right) = \sum_{j=1}^{\ell} \lambda_j T(\mathbf{x}_j).$$

In words, the function value of a linear combination is the linear combination of the function values.

Proof (with dots). We use the axioms (i) and (ii) of linear transformations in Definition 2.27 to obtain

$$T\left(\sum_{j=1}^{\ell}\lambda_j\mathbf{x}_j\right) = T\left(\sum_{j=1}^{\ell-1}\lambda_j\mathbf{x}_j + \lambda_\ell\mathbf{x}_\ell\right) \overset{(i)}{=} T\left(\sum_{j=1}^{\ell-1}\lambda_j\mathbf{x}_j\right) + T\left(\lambda_\ell\mathbf{x}_\ell\right) \overset{(ii)}{=} T\left(\sum_{j=1}^{\ell-1}\lambda_j\mathbf{x}_j\right) + \lambda_\ell T\left(\mathbf{x}_\ell\right).$$

Doing the same for $\ell-1$, we further get

$$T\left(\sum_{j=1}^{\ell}\lambda_j\mathbf{x}_j\right) = \underbrace{T\left(\sum_{j=1}^{\ell-2}\lambda_j\mathbf{x}_j\right) + \lambda_{\ell-1}T\left(\mathbf{x}_{\ell-1}\right)}_{T(\sum_{j=1}^{\ell-1}\lambda_j\mathbf{x}_j)} + \lambda_\ell T\left(\mathbf{x}_\ell\right).$$

Repeating this for $\ell-2,\ldots,1$, we finally obtain

$$T\left(\sum_{j=1}^{\ell}\lambda_j\mathbf{x}_j\right) = \underbrace{T\left(\sum_{j=1}^{0}\lambda_j\mathbf{x}_j\right)}_{T(\mathbf{0})} + \lambda_1 T(\mathbf{x}_1) + \ldots + \lambda_{\ell-1}T\left(\mathbf{x}_{\ell-1}\right) + \lambda_\ell T\left(\mathbf{x}_\ell\right) = \sum_{j=1}^{\ell}\lambda_j T(\mathbf{x}_j),$$

because $T(\mathbf{0}) = T(0\mathbf{x}) = 0T(\mathbf{x}) = \mathbf{0}$ for every $\mathbf{x}$, using axiom (ii). □

This proof has the usual dots in $\lambda_1 T(\mathbf{x}_1)+\cdots+\lambda_{\ell-1}T(\mathbf{x}_{\ell-1})+\lambda_\ell T(\mathbf{x}_\ell)$, but also dots of a different quality, namely the ones in $\ell-2,\ldots,1$. These dots indicate a repeating pattern in the proof itself. In Section 1.1.5, we have seen dot-free notations for sequences and sums and argued that they are more precise than the ones with the dots; is there also a dot-free notation for proofs such as the one above? Yes, and this notation is known as *proof by induction*, an important proof technique in mathematics. The concept of induction is the following: suppose we want to prove that some statement holds for all natural numbers n. For example, that

$$\sum_{j=1}^{n} j = \frac{n(n+1)}{2}.$$

We first check the *base case*, meaning that the statement holds for the first natural number $n=0$. Indeed, for $n=0$, both sides are 0.

For $n>0$, we perform the *induction step*. This proves an implication: *if* the statement holds for the number $n-1$ (this is the *induction hypothesis*), *then* it also holds for n (this concludes the induction step). In our example, if the statement holds for $n-1$, we can compute

$$\sum_{j=1}^{n} j = \left(\sum_{j=1}^{n-1} j\right) + n \underset{\substack{\uparrow\\ \text{induction hypothesis}}}{=} \frac{(n-1)n}{2} + n \underset{\substack{\uparrow\\ \text{easy calculation}}}{=} \frac{n(n+1)}{2}.$$

So if the statement holds for $n-1$, it indeed also holds for n.

Once we have completed base case and induction step, we have proved the statement for all natural numbers. Why is that? We have proved it for the first natural number (base case), and the induction step lets us conclude that it also holds for the second natural number (if it holds for the first, then it also holds for the second). So we have proved it for the second natural number, and the induction step lets us conclude that it also holds for the third natural number. And so on (these are the dots, but now they don't appear in the proof, but in its justification). Every natural number is eventually reached by this sequence of steps, so we have proved it for all natural numbers. Even though there are infinitely many natural numbers, every natural number itself is finite, so we eventually get to it.

Let's exercise induction for the statement of Lemma 2.28. Here, the natural number is not called n but ℓ.

Proof (by induction). Base case: For $\ell = 0$, the statement reads as $T(\mathbf{0}) = \mathbf{0}$ which we already saw to be true in the proof with dots. For $\ell > 0$, we perform the induction step: if the statement holds for $\ell - 1$, we compute

$$\begin{aligned} T\left(\sum_{j=1}^{\ell} \lambda_j \mathbf{x}_j\right) &= T\left(\sum_{j=1}^{\ell-1} \lambda_j \mathbf{x}_j + \lambda_\ell \mathbf{x}_\ell\right) \overset{(i)}{=} T\left(\sum_{j=1}^{\ell-1} \lambda_j \mathbf{x}_j\right) + T\left(\lambda_\ell \mathbf{x}_\ell\right) \\ &\overset{(ii)}{=} T\left(\sum_{j=1}^{\ell-1} \lambda_j \mathbf{x}_j\right) + \lambda_\ell T\left(\mathbf{x}_\ell\right) \\ &= \sum_{j=1}^{\ell-1} \lambda_j T(\mathbf{x}_j) + \lambda_\ell T\left(\mathbf{x}_\ell\right) \quad \text{(induction hypothesis)} \\ &= \sum_{j=1}^{\ell} \lambda_j T(\mathbf{x}_j). \end{aligned}$$

So if the statement holds for $\ell - 1$, it indeed holds for ℓ. □

The proof is conceptually the same as the previous one, but replaces "repeating this for $\ell - 2, \ldots, 1$" by a single step. Whenever you see a proof using repetitions, or saying "and so on...", you can be pretty sure that this an informal proof by induction. There is nothing wrong with a proof using "and so on", as long as you can turn it into a formal proof by induction, if needed.

Looking back at the proof of Lemma 2.10, this was also an induction in disguise, where we have used "one by one" instead of "and so on" to indicate a repeating step in a proof.

2.3.3 The matrix of a linear transformation

We have seen that every matrix A defines a linear transformation T_A (Definition 2.25). Now we can prove that *every* linear transformation $T : \mathbb{R}^n \to \mathbb{R}^m$ (Definition 2.27) is of the form $T = T_A$ for a unique $m \times n$ matrix A. This means, linear transformations are in some sense "the same" as matrices, but provide a different (and very useful) interpretation of matrices as functions that "do something" (transform input vectors to output vectors).

Theorem 2.29. *Let $T : \mathbb{R}^n \to \mathbb{R}^m$ be a linear transformation. There exists a unique $m \times n$ matrix A such that $T = T_A$.*

Proof. In order for $T = T_A$ to hold, we must have $T(\mathbf{e}_j) = T_A(\mathbf{e}_j) = A\mathbf{e}_j$ for all $j \in [n]$. Since $A\mathbf{e}_j$ is the j-th column of A, the only candidate for A is

$$A = \begin{bmatrix} | & | & & | \\ T(\mathbf{e}_1) & T(\mathbf{e}_2) & \cdots & T(\mathbf{e}_n) \\ | & | & & | \end{bmatrix},$$

the matrix whose columns are the m-dimensional output vectors that we get when we apply T to the n-dimensional standard unit vectors as inputs. With this matrix A, we indeed get

$$T_A(\mathbf{x}) = A\mathbf{x} = \sum_{j=1}^{n} x_j T(\mathbf{e}_j) = T\left(\sum_{j=1}^{n} x_j \mathbf{e}_j\right) = T(\mathbf{x}).$$

The first equality just applies Definition 2.25, in the second one we use Definition 2.4 of matrix-vector multiplication with our matrix A as defined above. In the third equality, we apply Lemma 2.28, and the last one uses that every vector $\mathbf{x}$ is a linear combination of the standard unit vectors, where the scalars are simply the entries of $\mathbf{x}$, as in

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_1 \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{e}_1} + x_2 \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}_{\mathbf{e}_2} + x_3 \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{e}_3}.$$

□

A consequence of the proof is the following: every linear transformation is completely determined by its behavior on the standard unit vectors. This also explains why we have paid special attention to this behavior in Figures 2.8 and 2.9.

2.3.4 Linear transformations and matrix multiplication

If you have so far thought that the definition of matrix multiplication (Section 2.2) is a bit artificial, linear transformations may change your mind about this. Let's say we have two linear transformations $T_A : \mathbb{R}^n \to \mathbb{R}^a$ and $T_B : \mathbb{R}^b \to \mathbb{R}^n$. Then we can define a new function $T : \mathbb{R}^b \to \mathbb{R}^a$ via

$$T(\underbrace{\mathbf{x}}_{\in\mathbb{R}^b}) = \underbrace{T_A(\underbrace{T_B(\mathbf{x})}_{\in\mathbb{R}^n})}_{\in\mathbb{R}^a},$$

i.e. we first apply T_B to $\mathbf{x}$ and then T_A to the result. This operation is known as the *composition* of functions.

It is easy to check that T is again a linear transformation, so we know from Theorem 2.29 that there is a matrix C such that $T = T_C$, but what is this matrix? The answer is that $C = AB$, the product of A and B!

Lemma 2.30. *Let $T_A : \mathbb{R}^n \to \mathbb{R}^a$ and $T_B : \mathbb{R}^b \to \mathbb{R}^n$ be two linear transformations. Then*

$$T_A(T_B(\mathbf{x})) = T_{AB}(\mathbf{x}) \textit{ for all } \mathbf{x} \in \mathbb{R}^b.$$

Proof. This is a one-liner:

$$T_A(T_B(\mathbf{x})) = T_A(B\mathbf{x}) = A(B\mathbf{x}) = (AB)\mathbf{x} = T_{AB}(\mathbf{x}).$$

Here, we have used Definition 2.25 for T_A, T_B, T_{AB} as well as associativity of matrix multiplication (Lemma 2.22), where we treat $\mathbf{x}$ as a $b \times 1$ matrix. □

Wit this, we can now also understand generalized associativity of matrix multiplication (Section 2.2.2). This is the fact that the placement of brackets doesn't matter in the product of several matrices; let's prove $(AB)(CD) = A((BC)D)$ as an example. Using Lemma 2.30 three times, we get that

$$T_{(AB)(CD)}(\mathbf{x}) = T_{AB}(T_{CD}(\mathbf{x})) = T_{AB}(T_C(T_D(\mathbf{x}))) = T_A(T_B(T_C(T_D(\mathbf{x})))).$$

The initial brackets in $(AB)(CD)$ have disappeared: to compute $T_{(AB)(CD)}(\mathbf{x})$, we simply apply T_D, T_C, T_B and T_A in this order.

Let's do the same for $A((BC)D)$. We get

$$T_{A((BC)D)}(\mathbf{x}) = T_A(T_{(BC)D}(\mathbf{x})) = T_A(T_{BC}(T_D(\mathbf{x}))) = T_A(T_B(T_C(T_D(\mathbf{x})))).$$

In both cases, the result is the same for all $\mathbf{x}$, so $T_{A((BC)D)}$ and $T_{A((BC)D)}$ are the same functions. With Theorem 2.29, we can conclude that also the matrices must be the same, hence $(AB)(CD) = A((BC)D)$.

We refrain from stating it as a formal theorem, but if we have a product P of matrices $A_1, A_2, \ldots, A_k$, with brackets put in an arbitrary manner, then $T_P(\mathbf{x})$ is computed by applying $T_{A_k}, T_{A_{k-1}}, \ldots, T_{A_1}$ in this order, so the result (and therefore also the product P) does not depend on where the brackets are.

2.3.5 Kernel and Image

For every linear transformation, there are two important sets of vectors.

Definition 2.31 (Kernel and image)**.** *Let $T : \mathbb{R}^n \to \mathbb{R}^m$ be a linear transformation. The set*

$$\mathbf{Ker}(T) := \{\mathbf{x} \in \mathbb{R}^n : T(\mathbf{x}) = \mathbf{0}\} \subseteq \mathbb{R}^n$$

is the kernel *of T. The set*

$$\mathbf{Im}(T) := \{T(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\} \subseteq \mathbb{R}^m$$

is the image *of T.*

$T : \mathbb{R}^n \to$	$T(\mathbf{x})$	$\mathbf{Ker}(T)$	$\mathbf{Im}(T)$
$\mathbb{R}^1$	$\sum_{i=1}^n x_i$	$\{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i = 0\}$	$\mathbb{R}$
$\mathbb{R}^m$	$\mathbf{0}$	$\mathbb{R}^n$	$\{\mathbf{0}\}$
$\mathbb{R}^n$	$\mathbf{x}$	$\{\mathbf{0}\}$	$\mathbb{R}^n$

Table 2.3: Kernel and image of some linear transformations

The image of T is the set of all outputs that T can produce. This is actually a familiar concept.

Observation 2.32. *Let $T : \mathbb{R}^n \to \mathbb{R}^m$ be a linear transformation and A the $m \times n$ matrix such that $T = T_A$ (see Theorem 2.29). Then*

$$\mathbf{Im}(T) = \mathbf{C}(A),$$

the column space of A.

This immediately follows from Definition 2.8 of $\mathbf{C}(A)$ and $T(\mathbf{x}) = A\mathbf{x}$. In light of this, some sources also call the column space of A the image of A.

Similarly, the kernel (the set of all inputs that produce output $\mathbf{0}$) can be expressed as

$$\mathbf{Ker}(T) := \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{0}\}.$$

This is a set that we will later call the *nullspace* of A (Definition 4.31). Again, some sources also call this the kernel of A.

Table 2.3 provides kernel and image of the linear transformations that we have considered as examples on page 60.

Exercise 2.33. *Let $\mathbf{v} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n$ be nonzero vectors and consider the $m \times n$ matrix $A = \mathbf{v}\mathbf{w}^\top$ (this matrix has rank 1 by Lemma 2.21). Give formulas for $\mathbf{Ker}(T_A)$ and $\mathbf{Im}(T_A)$, depending on the vectors $\mathbf{v}$ and $\mathbf{w}$!*

Chapter 3

Solving Linear Equations $A\mathrm{x} = \mathrm{b}$

3.1 Systems of linear equations

How to solve systems of linear equations is one of the foundational problems of linear algebra. Such systems appear in many applications today and can be very large. One concrete application that we present is Google's PageRank algorithm. We introduce systems of linear equations formally, using matrices and vectors, and we explain their "computer versions" that are used in computer programs for solving systems of linear equations.

We have already seen a system of linear equations in Section 0.3:

$$\begin{aligned} D &= 2S \\ D &= C + 3 \\ D + S + C &= 17 \end{aligned}$$

With its three equations in three variables, this system encodes three pieces of information about the ages of three children (**D**ominik, **S**usanne and **C**laudia). Solving the system means to find values for the variables such that all the equations are satisfied.

In general, systems of linear equations can have arbitrarily many equations and variables; for a systematic treatment, we write such systems in a standard form, and we use vectors and matrices to argue about them. In standard form, the variables are called $x_1, x_2, \ldots$ and appear only left of "=" in the equations. In this form, the system from the children's age puzzle is

$$\begin{aligned} x_1 - 2x_2 &= 0 \\ x_1 - x_3 &= 3 \\ x_1 + x_2 + x_3 &= 17 \end{aligned} \tag{3.1}$$

Here, x_1 stands for D, x_2 for S and x_3 for C.

Definition 3.1 (System of linear equations). *A system of linear equations in m equations and n variables $x_1, x_2, \ldots, x_n$ is of the form*

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ & \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & = & b_m, \end{array}$$

where the a_{ij} and b_i stand for known real numbers, and the x_i stand for unknown real numbers that we want to compute such that they satisfy all the equations. In matrix-vector form, this can be written as

$$A\mathbf{x} = \mathbf{b}: \quad \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}}_{A,\ m\times n} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}\in\mathbb{R}^n} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}}_{\mathbf{b}\in\mathbb{R}^m}.$$

(Here we use matrix-vector multiplication in the form of Observation 2.5.)A is the coefficient matrix, $\mathbf{b}$ *is the* right-hand side, *and* $\mathbf{x}$ *is the* vector of variables. *Solving the system means to compute a vector $\mathbf{x} \in \mathbb{R}^n$ such that $A\mathbf{x} = \mathbf{b}$.*

In the form $A\mathbf{x} = \mathbf{b}$, system (3.1) reads as

$$\underbrace{\begin{bmatrix} 1 & -2 & 0 \\ 1 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 \\ 3 \\ 17 \end{bmatrix}}_{\mathbf{b}}.$$

In this language, we can formulate linear independence of the columns of a matrix A in the following way. x

Observation 3.2. *Let A be an $m \times n$ matrix. The columns of A are linearly independent if and only if the system $A\mathbf{x} = \mathbf{0}$ has a unique solution, $\mathbf{x} = \mathbf{0}$.*

Proof. By Definition 2.4 of matrix-vector multiplication, the solutions are precisely the vectors of scalars that express $\mathbf{0}$ as a linear combination of the columns. A unique solution means that $\mathbf{0}$ can only be written as a trivial linear combination of the colums. By Lemma 1.19, this is equivalent to the columns of A being linearly independent. □

3.1.1 The PageRank algorithm

As an example where (large) systems of linear equations come up, we will discuss the PageRank algorithm. This algorithm was published by the Stanford students Sergey Brin and Lawrence Page in 1998 [BP98, pp. 109-110]. The first sentence of the abstract is the following:

> In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertexts.

The rest is history.

To understand what PageRank does, let's look at an example. Figure 3.1 shows a *link graph* where the circles represent web pages and the arrows indicate links between them. For the whole internet, you can think of a similar figure with a billion (10^9) circles.

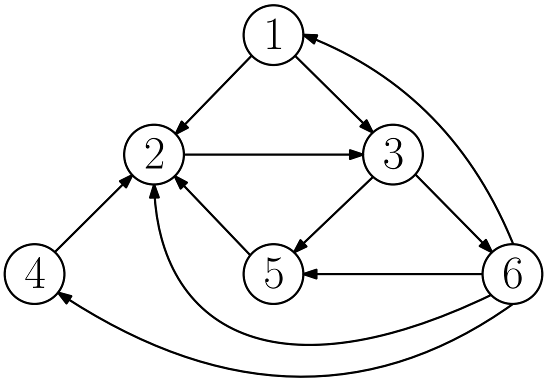


Figure 3.1: A link graph. The circles represent web pages, and an arrow from page p to page q indicates that p has a link to q. The PageRank algorithm sorts the pages by relevance.

Which of the six pages shown in Figure 3.1 do you think is most relevant? It's not clear what we mean by that, so let's start by measuring relevance in terms of the number of links pointing to a page. These are also called *citations* of the page. In scientific literature, the number of citations that a paper has is indeed an established measure of relevance, so it seems natural to also apply it to web pages.

This has been done long before PageRank, and according to this measure, page 2 (with 4 citations) is clearly the most relevant one; all other pages have at most 2 citations.

The main insight behind PageRank is that not all citations are worth the same. Here are the two key points.

1. Citations from relevant pages should count more than citations from irrelevant ones. It is too easy to collect many citations from irrelevant pages created for that purpose.

2. If a page cites a large number of other pages, an individual citation on that page should count less. It is too easy to cite many random pages.

In both points, web page citations fundamentally differ from scientific citations, where it is not so easy to manipulate citation counts in the described way (although it has become easier with predatory publishers that do not enforce quality standards).

Point 2 is easy to incorporate into the relevance measurement by simply giving less weight to citations from pages that cite many other pages. How to address point 1 is less clear. We want to define the relevance of a page depending on the relevances of the pages that cite it, but this definition goes in circles. In Figure 3.1, page 2 cites page 3 which cites page 5 which cites page 2.

A system of linear equations comes to the rescue. Let's focus on the situation in Figure 3.1 and introduce variables $x_1, x_2, \ldots, x_6$ for the relevances of pages $1, 2, \ldots, 6$.

We concretely define the relevance of a page as the sum of the *weighted* relevances of the pages that cite it, where the weighted relevance of a page is its relevance, divided by the number of citations on the page (this addresses point 2). Putting this in formulas for the relevance of page 2, we get the linear equation

$$x_2 = \frac{x_1}{2} + x_4 + x_5 + \frac{x_6}{4}.$$

We have contributions from each of the four pages that cite page 2. For page 6, the weighted relevance is only $x_6/4$, because page 6 cites 4 pages. Repeating this for all pages, we obtain a system of 6 equations in 6 variables.

Unfortunately, setting all the x_j to 0 is a solution from which we learn nothing. To avoid this, PageRank introduces a *damping factor* $0 < d < 1$ and replaces the equation for x_2 (and all others in the same way) by

$$x_2 = (1-d) + d\left(\frac{x_1}{2} + x_4 + x_5 + \frac{x_6}{4}\right).$$

For $d = 1$, we get the previous system with the useless all-zero solution, but for $d < 1$, it can be proved that the system has a unique solution with all page relevances summing up to the number of pages. Brin and Page suggest to use $d \approx 0.85$. Using $d = 7/8$, the relevances (which are then called page ranks) can be computed for example by pasting the following code into Wolfram Alpha.[1]

```
solve(
d=7/8,
x1=(1-d)+d*(x6/4),
x2=(1-d)+d*(x1/2+x4+x5+x6/4),
x3=(1-d)+d*(x1/2+x2),
x4=(1-d)+d*(x6/4),
x5=(1-d)+d*(x3/2+x6/4),
x6=(1-d)+d*(x3/2)
)
```

The solution (rounded to five digits) is

$$x_1 = 0.31797, x_2 = 1.6761, x_3 = 1.7307, x_4 = 0.31797, x_5 = 1.0751, x_6 = 0.88217.$$

This means, according to PageRank, page 3 is actually the most relevant one, followed by pages 2, 5, 6. Pages 1 and 4 are the least relevant ones, with the same low page rank.

How to solve systems of linear equations like that efficiently will be the focus of this chapter.

[1] https://www.wolframalpha.com/

3.1.2 Computer vectors and matrices

Systems of linear equations are solved by *algorithms*, stepwise procedures to solve a given problem. We explain the algorithms with examples and give mathematical proofs of correctness, but in one case, we also present the full algorithm in computer code that can directly be used in Java and C++ programs, for example.

For this, we first need to understand how matrices and vectors are represented in a computer program. While the details depend on the programming language being used, the concept is common to many languages. We introduce the concept here on a somewhat abstract level (but it directly works like that in Java and C++).

A vector such as the right-hand side $\mathbf{b}$ is represented by an *array*. For a vector $\mathbf{b} \in \mathbb{R}^m$, we use an array named $\mathtt{b}$ with entries $\mathtt{b}[0], \mathtt{b}[1], \ldots, \mathtt{b}[\mathtt{m}-1]$. We can visualize $\mathtt{b}$ as

$$\mathtt{b} = \begin{bmatrix} \mathtt{b}[0] \\ \mathtt{b}[1] \\ \vdots \\ \mathtt{b}[\mathtt{m}-1] \end{bmatrix}$$

and call it a *computer vector*. Array indices start from 0, not from 1, this is the main thing one needs to get used to with arrays. The reason for this is that an array typically occupies a contiguous chunk of computer memory, starting at the memory location of its first entry. The index of an entry tells us how many locations "further to the right" the entry is. As the first entry is 0 locations further to the right, it consequently has index 0.

A matrix such as the coefficient matrix A is represented by a *two-dimensional array*. This is an array of arrays. For a matrix $A \in \mathbb{R}^{m\times n}$, we use an array named $\mathtt{A}$ with m entries $\mathtt{A}[0], \mathtt{A}[1], \ldots, \mathtt{A}[\mathtt{m}-1]$ that are itself arrays with n entries each, representing the rows of A. This can be visualized as

$$\mathtt{A} = \begin{bmatrix} — & \mathtt{A}[0] & — \\ — & \mathtt{A}[1] & — \\ & \vdots & \\ — & \mathtt{A}[\mathtt{m}-1] & — \end{bmatrix}$$

and called a *computer matrix* in row notation. Sometimes, it makes sense to interpret the entries of $\mathtt{A}$ as the columns of A, but rows are better aligned with our understanding of a matrix entry a_{ij} as being in row i and column j. Indeed, if we visualize row $\mathtt{A}[\mathtt{i}]$ of our computer matrix as

$$\mathtt{A}[\mathtt{i}] = \begin{bmatrix} \mathtt{A}[\mathtt{i}][0] & \mathtt{A}[\mathtt{i}][1] & \cdots & \mathtt{A}[\mathtt{i}][\mathtt{n}-1] \end{bmatrix},$$

we get that $\mathtt{A}[\mathtt{i}][\mathtt{j}]$ is the entry of $\mathtt{A}$ in row i and column j (both counting from 0).

In a full computer program, the entries of computer vectors and matrices also have *types* that encode what kind of numbers they represent. To represent real numbers on a computer, one typically uses *floating-point numbers*, and the corresponding types tend to be called *`float`* and *`double`* (standing for double precision). But floating-point numbers only approximate real numbers, and the problems with that are studied in *numerical analysis*. Here, we abstract from this issue and do not talk about types. In writing down the algorithms, we simply pretend that entries of computer vectors and matrices are also real numbers.

3.2 Gauss elimination

We present Gauss elimination, the classical algorithm for solving systems of m equations in the same number m of variables. Our version does not always work, but is particularly simple and provides important insights into the structure of the coefficient matrix A. As we show, the algorithm works exactly when the matrix A has linearly independent columns. We will also count the number of steps that the algorithm performs.

For this section, we restrict to the case where the system $A\mathbf{x} = \mathbf{b}$ has a square coefficient matrix, i.e. A is assumed to be an $m \times m$ matrix. This is the important case of "m equations in m variables." For example, the PageRank algorithm described in Section 3.1.1 needs to solve a system of this kind. The general case of arbitrary A will be treated in Section 3.5.

3.2.1 Back substitution

We start with an easy case: if A is *upper triangular* (Definition 2.3 (iii)), the system $A\mathbf{x} = \mathbf{b}$ can be solved by *back substitution*. Let's look at a 3×3 example:

$$\begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 6 \\ 0 & 0 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 19 \\ 17 \\ 14 \end{bmatrix}.$$

Here, the blue entries are the ones that need to be zero in an upper triangular matrix. Due to this matrix shape, equation 3 has only variable x_3:

$$7x_3 = 14.$$

We can solve this directly and get $x_3 = 2$. Equation 2 has variables x_2 and x_3, but as we already know the value of x_3, we can substitute x_3 with this value and then directly solve for x_2. In equation 1, we finally have all three variables, but if we substitute x_2 and x_3 with their already known values, we can directly solve for x_1 and are done.

Table 3.1 summarizes the steps.

If A is an upper triangular $m \times m$ matrix, this works in the same way. We go through the equations in backwards order $m, m-1, \ldots, 1$. Equation i reads as

$$\sum_{j=i}^{m} a_{ij} x_j = b_i.$$

equation	before substitution	after substitution	solution
1	$2x_1 + 3x_2 + 4x_3 = 19$	$2x_1 + 11 = 19$	$x_1 = 4$
2	$5x_2 + 6x_3 = 17$	$5x_2 + 12 = 17$	$x_2 = 1$
3	$7x_3 = 14$		$x_3 = 2$

Table 3.1: Back substitution in an upper triangular system of 3 equations in 3 variables

We already know the values for $x_{i+1}, \ldots, x_m$ from the equations below. So we can substitute these variables with their values and then directly solve for x_i, giving us

$$x_i = \frac{b_i - \sum_{j=i+1}^{m} a_{ij}x_j}{a_{ii}}.$$

But this only works if the diagonal entry a_{ii} is nonzero, since we have to divide by it. This means, we need an upper triangular matrix where all diagonal entries are nonzero. Table 3.2 provides the algorithm in computer code. Note that the row and column indices are between 0 and $m-1$, in agreement with the convention on computer vectors and matrices that we have discussed in Section 3.1.2.

```
for (i = m-1; i >= 0; i--) {
    sum = b[i];
    for (j = i+1; j < m; j++)
        sum -= A[i][j] * x[j];
    x[i] = sum / A[i][i];
}
```

Table 3.2: The back substitution algorithm in computer code. The syntax is Java / C++.

3.2.2 Elimination

If the input matrix A is not upper triangular, *elimination* is trying to transform the system $A\mathbf{x} = \mathbf{b}$ into an equivalent system $U\mathbf{x} = \mathbf{c}$ where U is an upper triangular matrix. Equivalent means that both systems have the same solutions. If elimination succeeds, we can solve $U\mathbf{x} = \mathbf{c}$ using back substitution and obtain a solution of $A\mathbf{x} = \mathbf{b}$.

Elimination transforms the system step by step. Again, we demonstrate this with a 3×3 example where

$$A = \begin{bmatrix} 2 & 3 & 4 \\ 4 & 11 & 14 \\ 2 & 8 & 17 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 19 \\ 55 \\ 50 \end{bmatrix}. \tag{3.2}$$

We would like to turn the three red nonzero entries into zeros so that the matrix becomes upper triangular. We do this column by column, from left to right. To get rid of the 4, we subtract $2 \cdot$ (equation 1) from equation 2 of the system:

$$\begin{array}{rrcrcrcrcr} & \text{(equation 2)} & : & 4x_1 & + & 11x_2 & + & 14x_3 & = & 55 \\ - & 2\cdot\text{(equation 1)} & : & 4x_1 & + & 6x_2 & + & 8x_3 & = & 38 \\ \hline & \text{(equation 2')} & : & & & 5x_2 & + & 6x_3 & = & 17 \end{array}$$

This eliminates variable x_1 from the second equation, and we obtain an updated system $A'\mathbf{x} = \mathbf{b}'$ with a new second equation, given by

$$A' = \begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 6 \\ 2 & 8 & 17 \end{bmatrix}, \mathbf{b}' = \begin{bmatrix} 19 \\ 17 \\ 50 \end{bmatrix}.$$

In this step, $\mathbf{x}$ is just a distraction, all that matters are the entries of A and $\mathbf{b}$. To get A' from A, we subtract $2 \cdot$ (row 1) from (row 2). The same operation transforms $\mathbf{b}$ to $\mathbf{b}'$ (here, a row is just a single number).

Subtracting a multiple of some row from another row is a *row subtraction*. This can also be viewed as a linear transformation applied to all columns of A, and to $\mathbf{b}$. We further know that each linear transformation comes from a matrix; see Section 2.3.3. In our example, the transformation is

$$T_{E_{21}}\left(\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}\right) = \begin{bmatrix} v_1 \\ v_2 - 2v_1 \\ v_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{E_{21}} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}.$$

Hence, matrix and right-hand side of the transformed system $A'\mathbf{x} = \mathbf{b}'$ can be computed as

$$A' = E_{21}A, \quad \mathbf{b}' = E_{21}\mathbf{b}, \qquad E_{21}\text{: "subtract } 2\cdot\text{(row 1) from (row 2)".}$$

E_{21} is called an *elimination matrix*. Generally, we use E_{ij} to denote an elimination matrix that is supposed to create a zero entry in row i and column j.

To argue that this transformation does not change the solutions, we need that it can be undone:

$$A = E'_{21}A', \quad \mathbf{b} = E'_{21}\mathbf{b}', \qquad E'_{21}\text{: "add } 2\cdot\text{(row 1) to (row 2)".}$$

In the 3×3 case, the matrix for this *row addition* is

$$E'_{21} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Now it easy to see that the two systems $A\mathbf{x} = \mathbf{b}$ and $A'\mathbf{x} = \mathbf{b}'$ have the same solutions. First, if $A\mathbf{x} = \mathbf{b}$, then

$$A'\mathbf{x} = E_{21}A\mathbf{x} = E_{21}\mathbf{b} = \mathbf{b}'.$$

So every solution of $A\mathbf{x} = \mathbf{b}$ is also solution of $A'\mathbf{x} = \mathbf{b}'$. And if $A'\mathbf{x} = \mathbf{b}'$, then

$$A\mathbf{x} = E'_{21}A'\mathbf{x} = E'_{21}\mathbf{b}' = \mathbf{b}.$$

So every solution of $A'\mathbf{x} = \mathbf{b}'$ is also solution of $A\mathbf{x} = \mathbf{b}$.

Continuing from $A', \mathbf{b}'$, there are two more steps to turn the remaining two red entries into zeros; in each step, the diagonal entry of the current column is used to eliminate the nonzeros below it. This entry is called the *pivot*. Table 3.3 summarizes all three steps. The result is precisely the system in upper triangular form that we have previously solved with back substitution in Section 3.2.1. As solutions never change during elimination,

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

also solves the original system (3.2) (check this!).

fat number: the **pivot**	$A = \begin{bmatrix}\mathbf{2} & 3 & 4\\ 4 & 11 & 14\\ 2 & 8 & 17\end{bmatrix}$	$\mathbf{b} = \begin{bmatrix}19\\55\\50\end{bmatrix}$
subtract $2\cdot$(row 1) from (row 2):	$\downarrow$	$\downarrow$
$E_{21} = \begin{bmatrix}1 & 0 & 0\\ -2 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$	$E_{21}A = \begin{bmatrix}\mathbf{2} & 3 & 4\\ 0 & 5 & 6\\ 2 & 8 & 17\end{bmatrix}$	$E_{21}\mathbf{b} = \begin{bmatrix}19\\17\\50\end{bmatrix}$
subtract $1\cdot$(row 1) from (row 3):	$\downarrow$	$\downarrow$
$E_{31} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -1 & 0 & 1\end{bmatrix}$	$E_{31}E_{21}A = \begin{bmatrix}2 & 3 & 4\\ 0 & \mathbf{5} & 6\\ 0 & 5 & 13\end{bmatrix}$	$E_{31}E_{21}\mathbf{b} = \begin{bmatrix}19\\17\\31\end{bmatrix}$
subtract $1\cdot$(row 2) from (row 3):	$\downarrow$	$\downarrow$
$E_{32} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -1 & 1\end{bmatrix}$	$\underbrace{E_{32}E_{31}E_{21}A}_{U} = \begin{bmatrix}2 & 3 & 4\\ 0 & 5 & 6\\ 0 & 0 & \mathbf{7}\end{bmatrix}$	$\underbrace{E_{32}E_{31}E_{21}\mathbf{b}}_{\mathbf{c}} = \begin{bmatrix}19\\17\\14\end{bmatrix}$
↑ elimination matrices	done!	

Table 3.3: Elimination reduces a system of linear equations to upper triangular form. The red entries are the ones that must become 0 (blue entries).

Row exchanges. In a less nice case, it can happen that the pivot is 0 which we cannot use to eliminate nonzeros below it. But if there is some nonzero entry anywhere below the pivot, we can perform a *row exchange* to obtain a nonzero pivot. Table 3.4 gives an example for this situation. Here, we are immediately done after the row exchange, but in general, we would now use the new nonzero pivot to eliminate the nonzeros below.

$$A = \begin{bmatrix} \mathbf{2} & 3 & 4 \\ 4 & 6 & 14 \\ 2 & 8 & 17 \end{bmatrix} \qquad \mathbf{b} = \cdots$$

elimination in first column: $\downarrow$ $\downarrow$

$$E_{31}E_{21}A = \begin{bmatrix} 2 & 3 & 4 \\ 0 & \mathbf{0} & 6 \\ 0 & 5 & 13 \end{bmatrix} \qquad E_{31}E_{21}\mathbf{b} = \cdots$$

pivot **0**: exchange (row 2) and (row 3): $\downarrow$ $\downarrow$

$$P_{23} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \Bigg| \quad \underbrace{P_{23}E_{31}E_{21}A}_{U} = \begin{bmatrix} 2 & 3 & 4 \\ 0 & \mathbf{5} & 13 \\ 0 & 0 & 6 \end{bmatrix} \qquad \underbrace{P_{23}E_{31}E_{21}\mathbf{b}}_{\mathbf{c}} = \cdots$$

↑ permutation matrix

done!

Table 3.4: Elimination with row exchanges to ensure nonzero pivots

A row exchange can also be interpreted as a linear transformation; its matrix is a special *permutation matrix*. In general, a permutation matrix corresponds to a *permutation*, a linear transformation that reorders the entries of its input vector. A row exchange is a special permutation that only exchanges two entries. As before with row subtractions, we can argue that a row exchange does not change the solution of the linear system of equations. Here, this is even more obvious, as a row exchange simply corresponds to exchanging the order of equations.

Failure. Finally, there is an ugly case: the pivot is 0, and all entries below it are also 0, so that no row exchange helps and we are stuck. Table 3.5 shows an example. In this case, we give up for now. We also consider it ugly if this happens in the last column; see the right example in Table 3.5.

$$A = \begin{bmatrix} \mathbf{2} & 3 & 4 \\ 4 & 6 & 14 \\ 2 & 3 & 17 \end{bmatrix} \qquad \mathbf{b} = \cdots$$

elimination in first column: $\downarrow$ $\downarrow$

$$E_{31}E_{21}A = \begin{bmatrix} 2 & 3 & 4 \\ 0 & \mathbf{0} & 6 \\ 0 & 0 & 13 \end{bmatrix} \qquad E_{31}E_{21}\mathbf{b} = \cdots$$

no row exchange helps, give up for now!

$$\begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 6 \\ 0 & 0 & \mathbf{0} \end{bmatrix}$$

we also fail here!

Table 3.5: Elimination fails: no row exchange can ensure a nonzero pivot.

From a purely functional point of view, there would be no reason to give up. After all, the goal of the elimination steps is to remove the zeros below the diagonal; if in some column, we start with only zeros below the diagonal, we could consider this column done and move on to the next one. Indeed, this works and transforms *any* square matrix A to an upper triangular matrix U, possibly with some 0's on the diagonal. Back substitution becomes a bit more tricky, then (there might be no solutions, or many solutions of $U\mathbf{x} = \mathbf{c}$), but this can be handled. In many sources, Gauss elimination is presented like that for all matrices A, even for non-square ones.

When we give up, this is not laziness. Distinguishing success and failure in Gauss elimination will allow us to exactly understand the "good" square matrices, the ones for which we succeed. This kind of understanding provides important insights, so we prefer to call it (mathematically) lazy to run a version of Gauss elimination that always succeeds. But once we have understood success and failure, we will make sure to also develop methods that always succeed; see Section 3.5.

Gauss elimination in computer code. The code for Gauss elimination (with row subtractions and row exchanges) is given in Table 3.6. The code transforms A and $\mathbf{b}$ *in place*, meaning that the entries of the arrays `A` and `b` are being changed so that they in the end correspond to an equivalent upper triangular system.

The code corresponds to the previous discussion, but there are two points to notice: in subtracting a multiple of the pivot row j from some row i below, we only update the entries in columns $j, j+1, \ldots, m$ of row i: in all columns further to the left, there is nothing to do, since row j has zero entries there, created in previous elimination steps. Also, the entry `A[i][j]` becomes zero by design, so there is no need to compute it. The second point is that elimination in the last column does not perform any actual work, but still checks whether the entry in the lower right corner of the matrix is nonzero. At this point, we catch the ugly corner case mentioned in Table 3.5 (right).

3.2.3 Success and failure

In the previous section, we have seen that Gauss elimination may have to give up in solving a system of m linear equations in m variables. This seems unsatisfactory, and in reading other sources, you find variants of Gauss elimination that always succeed.

Here, we do not want to fix the algorithm but understand why it is broken. The insights gained through this will also take us further along in the theory of linear algebra. So we rather like to think of giving up as a *productive failure* [Kap14].

We have argued (in the examples) that each step of Gauss elimination transforms the current system of linear equations into another one with the same solutions. This property is very important to guarantee that by solving the upper triangular system that we get in the end, we also solve the original system.

Now, we want to make this argument in general, for systems of m equations in n variables. In this case, the elimination matrix corresponding to a row substraction is an

```
for (j = 0; j < m; j++) {
    // eliminate in column j
    if (A[j][j] == 0) {
        // zero pivot, try row exchange
        k = j+1;
        while (k < m && (A[k][j] == 0)) k++;
        if (k == m)
            return false; // no row exchange is possible, give up
        else {
            // exchange rows j and k ...
            row_A = A[j]; A[j] = A[k]; A[k] = row_A; // ... of A
            row_b = b[j]; b[j] = b[k]; b[k] = row_b; // ... of b
        }
    }
    // create zeros below A[j][j]
    for (i = j+1; i < m; i++) {
        // subtract c * row j from row i ...
        c = A[i][j] / A[j][j];
       A[i][j] = 0;
        for (k = j+1; k < m; k++)
            A[i][k] -= c * A[j][k]; // ... of A
        b[i] -= c * b[j];           // ... of b
    }
}
return true;
```

Table 3.6: The Gauss elimination algorithm in computer code. The return value indicates whether the elimination was successful.

$m \times m$ matrix E_{ij} of the form

$$E_{ij} = \begin{array}{cc} \left[\begin{array}{ccccc} \diagdown & & & & \\ & 1 & & & \\ & & \diagdown & & \\ & -c & & 1 & \\ & & & & \diagdown \end{array}\right] & \begin{array}{c} \\ \leftarrow j \\ \\ \leftarrow i \\ \\ \end{array} \\ \begin{array}{ccccc} & \uparrow & & \uparrow & \\ & j & & i & \end{array} & \end{array}$$

Here $\diagdown$ stands for a contiguous sequence of diagonal $1'$s, and all omitted entries are 0. Hence, E_{ij} is the identity matrix with an extra $-c$ in row i and column j. You can convince yourself (using the rules of matrix-vector multiplication; see Section 2.1.1) that this is indeed the matrix of the linear transformation "subtract $c\cdot$(row j) from (row i)." Applying it to all columns of A and to $\mathbf{b}$, we get a new system $A'\mathbf{x} = \mathbf{b}'$ with $A' = E_{ij}A, \mathbf{b}' = E_{ij}\mathbf{b}$.

A permutation matrix as it appears in Gauss elimination for a row exchange has the form

$$P_{jk} = \begin{array}{cl} \left[\begin{array}{ccccc} \ddots & & & & \\ & 0 & & 1 & \\ & & \ddots & & \\ & 1 & & 0 & \\ & & & & \ddots \end{array}\right] & \begin{array}{l} \\ \leftarrow j \\ \\ \leftarrow k \\ \\ \end{array} \\ \begin{array}{ccccc} & \uparrow & & \uparrow & \\ & j & & k & \end{array} & \end{array}$$

This is the matrix of the linear transformation "exchange (row j) and (row k)," and if we apply this row exchange, we get a new system $A'\mathbf{x} = \mathbf{b}'$ with $A' = P_{jk}A, \mathbf{b}' = P_{jk}\mathbf{b}$.

A *row operation* is either a row subtraction, or a row exchange, and the matrix of a row operation is called the *row operation matrix*. We can now prove the following result.

Lemma 3.3. *Let $A\mathbf{x} = \mathbf{b}$ be a system of m linear equations in n variables, and let $M \in \mathbb{R}^{m\times m}$ be a row operation matrix. Let $A' = MA$ and $\mathbf{b}' = M\mathbf{b}$ be the result of applying the row operation to both A and $\mathbf{b}$. Then the two systems $A\mathbf{x} = \mathbf{b}$ and $A'\mathbf{x} = \mathbf{b}'$ have the same solutions.*

Proof. This works as in the 3×3 examples: If $A\mathbf{x} = \mathbf{b}$, then

$$A'\mathbf{x} = MA\mathbf{x} = M\mathbf{b} = \mathbf{b}'.$$

For the other direction, let M' be the matrix of the row operation that undoes M, i.e. transforms A' and $\mathbf{b}'$ back to A and $\mathbf{b}$, by either adding back $c \cdot$ (row j) to (row i), or by exchanging row j and row k again. Now, if $A'\mathbf{x} = \mathbf{b}'$, then

$$A\mathbf{x} = M'A'\mathbf{x} = M'\mathbf{b}' = \mathbf{b}.$$

Hence, every solution of $A\mathbf{x} = \mathbf{b}$ is a solution of $A'\mathbf{x} = \mathbf{b}'$ and vice versa. □

Corollary 3.4. *Let A be an $m \times n$ matrix, let $M \in \mathbb{R}^{m\times m}$ be a row operation matrix, and let $A' = MA$ be the result of applying the row operation to A. Then A has linearly independent columns if and only if A' has linearly independent columns.*

Proof. We apply Lemma 3.3 with $\mathbf{b} = \mathbf{0}$ (and hence $\mathbf{b}' = \mathbf{0}$ as well). This gives us that $A\mathbf{x} = \mathbf{0}$ and $A'\mathbf{x} = \mathbf{0}$ have the same solutions. If there is just one solution, namely the zero vector, we get linearly independent columns in both cases. If there is also another solution, we get linearly dependent columns in both cases; see Observation 3.2. □

Now we can understand exactly when Gauss elimination succeeds.

Theorem 3.5. *Let $A\mathbf{x} = \mathbf{b}$ be a system of m linear equations in m variables. The following two statements are eqivalent.*

(i) Gauss elimination as in Table 3.6 succeeds.

(ii) The columns of A are linearly independent.

We will prove (i)⇒(ii) "normally", but (ii)⇒(i) ("if (ii), then (i)") is easier to prove by *contraposition*. This proves the logically equivalent implication "if *not* (i), then *not* (ii)". You know this from natural language. Consider the sentence "If it rains, then the street is wet." Another (logically equivalent) way of saying this is "If the street is not wet, then it does not rain". The symbol for not (logical *negation*) is ¬, so the contraposition can be written as ¬(i)⇒ ¬(ii).

Proof. (i) ⇒(ii): If Gauss elimination succeeds, it produces an upper triangular matrix U with all diagonal entries u_{jj} nonzero. Such a matrix U has linearly independent columns: no column is a linear combination of the previous ones, due to the nonzero diagonal elements (and zeros to the left of them). Recall Corollary 1.20 (iii) for this alternative definition of linear independence. Hence, also A has linearly independent columns by Corollary 3.4 (applied throughout all elimination steps).

¬(i)⇒ ¬(ii): If Gauss elimination fails in column j, we have an intermediate matrix A' with zeros in rows $j, j+1, \ldots, m$ of column j. But in all previous columns, elimination succeeded, and the diagonal entries are nonzero. Hence, A' looks like this:

$$A' = \left[\begin{array}{c|c|c} U & \mathbf{v} & \cdots \\ \hline 0 & \mathbf{0} & \cdots \end{array}\right], \quad U \in \mathbb{R}^{(j-1)\times(j-1)} \text{ upper triangular, all } u_{\ell\ell} \neq 0, \mathbf{v} \in \mathbb{R}^{j-1}.$$

From this, we construct a nonzero solution $\mathbf{x}$ to $A'\mathbf{x} = \mathbf{0}$, showing that A' (and hence also the original A, by repeated application of Corollary 3.4) has linearly dependent columns. We start by setting $x_{j+1}, x_{j+2}, \ldots, x_m = 0$.

This already ensures that the vector $A'\mathbf{x}$ has zeros in rows $j, j+1, \ldots, m$, and to get $A'\mathbf{x} = \mathbf{0}$, we also need zeros in the first $j-1$ rows, meaning that

$$U \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{j-1} \end{bmatrix}}_{\mathbf{y}} + x_j \mathbf{v} = \mathbf{0}.$$

This can be achieved for example by setting $x_j = -1$ and solving $U\mathbf{y} = \mathbf{v}$ by back substitution; see Section 3.2.1.

□

You may wonder whether this proof also works (or how it is to be interpreted) if Gauss elimination already fails in the first column ($j = 1$). In this case, the complete first column of the original matrix A is $\mathbf{0}$, and hence, the columns are linearly dependent for obvious reasons; see also Table 1.3 and the discussion before it.

3.2.4 Runtime

Whenever computer scientists see an algorithm, they ask how efficient it is. For any given problem, there are different algorithms, and they may differ in their efficiency. Naturally, we would like to identify the most efficient one. But in order to do so, we must first define and measure the efficiency of an algorithm. The most important measures are runtime and memory consumption. Here, we focus on runtime. But how do we measure this?

Suppose we are given a concrete system of linear equations, for example the one on page 70 that we previously fed into Wolfram Alpha to compute page ranks for the "toy internet" of Figure 3.1.

Now we know how we can solve it ourselves, using Gauss elimination with back substitution as an algorithm, as realized for example by the lines of code in Tables 3.6 and 3.2.

If we integrate these lines of code into a concrete computer program and run it on a concrete computer, we can measure the time it takes to solve our concrete system of linear equations. What we get is a number (in milliseconds, for example). On a different system of linear equations, and on a different computer, we will get a different number. So this kind of measurement is not very informative.

What we do instead is the following: we inspect the algorithm, identify the crucial steps, and try to count how many of them the algorithm makes, depending on m. This results in a function $g : \mathbb{N} \to \mathbb{N}$, where $g(m)$ is the number of crucial steps needed to solve a system with m linear equations in m variables. If the "crucial steps" account for most of the steps in the algorithm, $g(m)$ is also a good estimate for the algorithm's efficiency in terms of the total number of steps. This measure is independent of the computer that we will actually use to run the algorithm.

Here, the crucial steps are the *arithmetic steps*: how often does the algorithm add, subtract, multiply, or divide two numbers? On top of this, there are clearly other steps, but only rather simple ones, and not many more than arithmetic steps. For example, in subtracting a multiple of a row from another row (lines 20–22 in Table 3.6), the algorithm performs $(m - j)$ multiplications and $(m - j)$ subtractions. On top of this, there are extra steps: initialize a loop variable, check whether the loop is done, change a loop variable, change an array entry. But each of these extra steps also happens at most $m - j$ times.

Generally, we want to argue that for each arithmetic step, only a couple of extra steps are needed. This makes sense, since the main task is computation, and everything else should only be there to support the main task. To formalize this, we can imagine each step to cost CHF 1 each. The arithmetic steps have to pay not only for themelves, but also bear the cost of the extra steps. If we can charge the extra steps to the arithmetic steps in such a way that each arithmetic step pays at most CHF 10, say, then we know that the total number of steps is at most 10 times the number of arithmetic steps.

Ideally, we want to charge each extra step to the arithmetic step that it directly supports, and in this way (hopefully) never overcharge any single arithmetic step. For the extra steps in the aforementioned loop (lines 20-22), this is easy: we charge them directly to the multiplication that happens in the corresponding iteration of the loop.

The precise details of this kind of charging may be a little tricky, but it can be done in the same spirit for all extra steps, with one major exception: an unsuccessful search for a row exchange (lines 3-13) cannot easily be charged to any arithmetic step. For example, if A's first column is $\mathbf{0}$, Gauss elimination does not do a single arithmetic step, but still needs extra steps before it gives up. In this case, we have no one to pay for them, but the cost of these steps is small.

What we find in the end is the following: Let $g(m)$ be the number of arithmetic steps necessary to solve a system of m equations with m variables using Gauss elimination with back substitution. Furthermore, let $t(m)$ be the total number of steps needed for that in the worst case (we don't even have to be super precise about what we count as a step). Then there is a constant $c \in \mathbb{N}$ such that

$$t(m) \leq c(g(m) + m) \text{ for all } m \in \mathbb{N}. \tag{3.3}$$

Here, the "$+m$" pays for the few steps in an unsuccessful search for a row exchange that cannot be charged to any arithmetic steps. You can also frame this as adding m "ghost" arithmetic steps, after which we can make sure that every arithmetic step will be charged at most CHF c.

The previous inequality is our interpretation of the arithmetic steps accounting "for most" of the steps. If you want a less generous interpretation of "at most", you also need to upgrade other steps to crucial status, but this usually entails a more complicated analysis.

To summarize: for Gauss elimination with back substitution, we will find a function g such that $g(m)$ counts the number of arithmetic steps necessary to solve a system of m equations with m variables. After adding m, the total number of steps is by some constant factor c larger.

The precise value of c is difficult to determine. It depends on our charging scheme and on what we count as an "extra step". Therefore, it is common practice not to talk about c at all. Instead, we will say that the runtime (in terms of total step count) for input size m is

$$O(g(m) + m).$$

This *big-O-notation* means "at most $g(m) + m$, multiplied by some constant factor" (that does not depend on m but could otherwise be anything; for example, 10, or 100).

The good thing is that $O(g(m)+m)$ is also a valid description of the algorithm's actual runtime on any given computer. This is because the actual runtime (in whatever unit of time) is also just the total number of steps, multiplied by another constant that bounds the actual runtime of a single step.

After having set the stage, we can now formulate and prove the main result about the efficiency of Gauss elimination. Our count for the number of arithmetic steps in Theorem 3.6 will be precise in the case where elimination succeeds; otherwise, it is an upper bound. Generally, it is rare that we can count the steps of an algorithm precisely. But upper bounds "close to the truth" are usually all that is needed, in particular in an analysis involving a "big O."

Theorem 3.6. *Let $A\mathbf{x} = \mathbf{b}$ be a system of m linear equations in m variables, $m \geq 1$. Gauss elimination with back substitution solves $A\mathbf{x} = \mathbf{b}$ (or gives up) with at most*

$$g(m) = \frac{2}{3}m^3 + \frac{3}{2}m^2 - \frac{7}{6}m$$

arithmetic steps and therefore in time $O(m^3)$.

Here, $O(m^3)$ is a shorthand for $O(f(m))$ where $f(m) = m^3$.

Proof. Time $O(m^3)$ follows from the value of $g(m)$, since the first term $\frac{2}{3}m^3$ is the dominating one. We can for example say that that

$$g(m) + m \leq \frac{2}{3}m^3 + \frac{3}{2}m^3 = \frac{13}{6}m^3.$$

As argued before, the runtime is at most $c(g(m) + m)$ for some constant c, hence at most $c \cdot \frac{13}{6}m^3$ which is $O(m^3)$.

Let's count the number of arithmetic steps. For elimination, this can be done by inspecting the code in Table 3.6. Arithmetic steps happen in line 18 (1 division), line 21 (1 multiplication, 1 subtraction) and line 22 (1 multiplication, 1 subtraction). These are done repeatedly through the surrounding loops starting in lines 20 (`for (k...)`), 16 (`for (i...)`) and 1 (`for (j...)`). Taking the loop ranges into account, we arrive at the following total number of arithmetic steps (if we give up on the way, we have less):

$$e(m) = \underbrace{\sum_{j=0}^{m-1} \underbrace{\sum_{i=j+1}^{m-1} \left(\overbrace{1}^{\text{Line 18}} + \underbrace{\sum_{k=j+1}^{m-1} \overbrace{2}^{\text{Line 21}}}_{\text{Line 20}} + \overbrace{2}^{\text{Line 22}} \right)}_{\text{Line 16}}}_{\text{Line 1}}.$$

This triple sum looks a bit scary, but it's not too bad. The number in the big bracket is easily computed, and the sum over i as well:

$$\begin{aligned} e(m) = \sum_{j=0}^{m-1} \sum_{i=j+1}^{m-1} (2(m-j)+1) &= \sum_{j=0}^{m-1} (m-j-1)(2(m-j)+1) \\ &= \underbrace{2\sum_{j=0}^{m-1} (m-j-1)(m-j)}_{e_{\text{mul}}(m) = e_{\text{sub}}(m)} + \underbrace{\sum_{j=0}^{m-1} (m-j-1)}_{e_{\text{div}}(m)}. \end{aligned}$$

Here, $e_{\text{mul}}(m)$ counts the number of multiplications and $e_{\text{sub}}(m)$ the number of subtractions (both numbers are the same); $e_{\text{div}}(m)$ counts the number of divisions. After substituting $m - j - 1$ with ℓ, these become standard summations that can be found in collections of formulas (or proved by induction):

$$\begin{aligned} e_{\text{mul}}(m) = e_{\text{sub}}(m) &= \textstyle\sum_{\ell=0}^{m-1} \ell(\ell+1) &&= \tfrac{1}{3}(m^3 - m), \\ e_{\text{div}}(m) &= \textstyle\sum_{\ell=0}^{m-1} \ell &&= \tfrac{1}{2}(m^2 - m). \end{aligned}$$

For back substitution as in Table 3.2, we can count in the same way and obtain that the number of arithmetic steps is

$$\begin{aligned} b(m) = \underbrace{\sum_{i=0}^{m-1}\left(\underbrace{\sum_{j=i+1}^{m-1}\overbrace{2}^{\text{Line 4}}}_{\text{Line 3}} + \overbrace{1}^{\text{Line 5}}\right)}_{\text{Line 1}} &= \sum_{i=0}^{m-1}(2(m-i-1)+1) \\ &= 2\underbrace{\sum_{i=0}^{m-1}(m-i-1)}_{b_{\text{mul}}(m)=b_{\text{sub}}(m)} + \underbrace{\sum_{i=0}^{m-1} 1}_{b_{\text{div}}(m)}. \end{aligned}$$

This gives

$$\begin{aligned} b_{\text{mul}}(m) = b_{\text{sub}}(m) &= \textstyle\sum_{\ell=0}^{m-1} \ell &&= \tfrac{1}{2}(m^2 - m) \\ b_{\text{div}}(m) &= \textstyle\sum_{i=0}^{m-1} 1 &&= m. \end{aligned}$$

In summary, we get

$$\begin{aligned} e(m) &= 2e_{\text{mul}}(m) + e_{\text{div}}(m) &&= \tfrac{2}{3}(m^3 - m) + \tfrac{1}{2}(m^2 - m), \\ b(m) &= 2b_{\text{mul}}(m) + b_{\text{div}}(m) &&= (m^2 - m) + m, \\ g(m) &= e(m) + b(m) &&= \tfrac{2}{3}(m^3 - m) + \tfrac{3}{2}(m^2 - m) + m. \end{aligned}$$

After collecting the linear terms (the ones with "m"), we get the claimed value. □

Let us look at some concrete values of $g(m)$. For example, we get

$$g(1) = \frac{2}{3} + \frac{3}{2} - \frac{7}{6} = 1.$$

This corresponds to the fact that one linear equation $ax = b$ in one variable x can be solved with one arithmetic step (a division), resulting in $x = b/a$.

Here are some more values where we also list the dominating term $\frac{2}{3}m^3$ (rounded to integer) for comparison.

m	$g(m)$	$\frac{2}{3}m^3$, rounded to integer
2	9	5
3	28	18
4	62	43
5	115	83
6	191	144
7	294	229
8	428	341
9	597	486
$\vdots$	$\vdots$	$\vdots$
100	681550	666667
$\vdots$	$\vdots$	$\vdots$
1000	668165500	666666667

This shows that for larger systems, the step count becomes quite high, and the dominating term actually accounts for most of it. So

$$\frac{2}{3}m^3$$

is a pretty good lower bound for the number of arithmetic steps. From this, we can already conclude that Gauss elimination will be very slow for $m = 10^6$ (1 million) on a normal computer, even if that computer can perform an astonishing number of 10^{12} (1 trillion) arithmetic steps per second. Indeed, for $m = 10^6$, Gauss elimination needs at least $\frac{2}{3}m^3 = \frac{2}{3}10^{18}$ arithmetic steps, and these alone will take at least $\frac{2}{3}10^6$ seconds which is roughly a week. And if the system has $m = 10^7$ (10 million), the runtime goes up by factor of 1000, resulting in at least 20 years of computing.

In practice, systems with $m = 10^6$ and even larger are common and can be solved much faster by algorithms exploiting that typical constraint matrices A are *sparse*, meaning that most of their entries are 0. In such cases, we can use algorithms that only deal with the few nonzero entries, resulting in significantly less (arithmetic) steps.

3.3 Inverse matrices

Based on understanding success and failure of Gauss elimination, we look at the important class of invertible matrices, the square matrices that lead to an "undoable" linear transformation $\mathbf{x} \to A\mathbf{x}$. It turns out that these are exactly the ones with linearly independent columns, and therefore exactly the ones for which Gauss elimination works and computes the unique solution of $A\mathbf{x} = \mathbf{b}$.

In the proof of Lemma 3.3 (about the effect of a row operation during Gauss elimination), we have used the "inverse" row operation with matrix M', undoing a previous row operation with matrix M: if $MA = A'$, then $M'A' = A$. We can also write this as

$$M'MA = A,$$

and as this works for every $m \times n$ matrix A, we can set $A = I$, the identity matrix, to get (using Corollary 2.20 in the first equality) that

$$M'M = M'MI = I.$$

Hence, if we premultiply the matrix of the row operation with the matrix of the inverse row operation, we get the identity matrix. For example,

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{add } 2\cdot\text{(row 1)} \\ \text{to (row 2)}}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{subtract } 2\cdot\text{(row 1)} \\ \text{from (row 2)}}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{I}.$$

This is not surprising if you think of the row subtraction and row addition as linear transformations T_M and $T_{M'}$ whose effects cancel out (yield the identity transformation) when you apply them one after the other: $T_{M'}(T_M(\mathbf{x})) = T_I(\mathbf{x}) = \mathbf{x}$; see Section 2.3.4.

Here, we want to investigate this for general matrices M, not only the ones corresponding to row operations.

3.3.1 Definition and basic properties

Definition 3.7 (Invertible matrix). *Let M be an $m \times m$ matrix. M is called* invertible *if there exists an $m \times m$ matrix M^{-1} (called the* inverse *of M) such that*

$$MM^{-1} = M^{-1}M = I.$$

For a row operation matrix M, the matrix M' of the inverse row operation is indeed the inverse matrix according to this definition. We have already argued that $M'M = I$, but $MM' = I$ also holds, since the original row operation is the inverse of the inverse row operation. It is generally true for two $m \times m$ matrices A and B that $AB = I$ implies $BA = I$, so the first equality in Definition 3.7 is actually superfluous. However, we will only prove this in Exercise 3.12 below, so for the time being, we work with the "foolproof" Definition 3.7 of the inverse.

Let us look at the situation for $m = 1, 2$. For $m = 1$, an $m \times m$ matrix is just a number (between square brackets), the identity matrix is the number 1, and the inverse is the reciprocal number:

Case 1×1**.**

$$M = \begin{bmatrix} a \end{bmatrix} \quad \Rightarrow \quad M^{-1} = \begin{bmatrix} \frac{1}{a} \end{bmatrix} \quad \text{(if } a \neq 0\text{).}$$

Hence, the inverse exists unless $a = 0$.

Case 2×2**.**

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad \Rightarrow \quad M^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad \text{(if } ad - bc \neq 0\text{).}$$

To see that this formula for M^{-1} is correct, we simply check that it satisfies Definition 3.7. Again, we see that the inverse does not always exist, but the condition here ($ad - bc \neq 0$) is less obvious than for $m = 1$. It is not hard to see (try to see this!) that this condition—expressed in words—reads as follows: The two columns of M are linearly independent.

Could there also be a different inverse? No, if a matrix has an inverse, it is unique:

Lemma 3.8. *Let M be an $m \times m$ matrix with two inverses A and B. Then $A = B$.*

Proof. Using associativity of matrix multiplication (Lemma 2.22) as well as Corollary 2.20, we compute

$$A = IA = (BM)A = B(MA) = BI = B.$$

□

For $m \times m$ matrices with $m > 2$, there is also a formula for the inverse, but this requires the concept of *determinants* to which we will only get in the second part of the lecture. The next lemma lets us compute the inverse of a *product* of two invertible matrices.

Lemma 3.9. *Let A and B be invertible $m \times m$ matrices. Then AB is also invertible, and*

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

Recall that AB is the matrix of the linear transformation $T_{AB}(\mathbf{x}) = T_A(T_B(\mathbf{x}))$ (first apply T_B, then T_A); see Lemma 2.30. To undo this, we need to apply the inverse transformations in reverse order: $T_{B^{-1}}(T_{A^{-1}}(\mathbf{x})) = T_{B^{-1}A^{-1}}(\mathbf{x})$ (first undo T_A, then undo T_B). You can already consider this a proof sketch of the lemma, but there is also a more direct proof not involving linear transformations.

Proof.

$$(AB)(B^{-1}A^{-1}) = A(BB^{-1})A^{-1} = AIA^{-1} = AA^{-1} = I$$

and

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

□

This naturally extends to more matrices, for example $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$, but we skip the formal statement and its proof.

Inversion also commutes with transposition (Definition 2.11).

Lemma 3.10. *Let A be an invertible $m \times m$ matrix. Then the transpose matrix $A^\top$ is also invertible, and*

$$\left(A^\top\right)^{-1} = \left(A^{-1}\right)^\top .$$

Proof. We need to check that

$$\underbrace{A^\top \left(A^{-1}\right)^\top}_{(A^{-1}A)^\top} = \underbrace{\left(A^{-1}\right)^\top A^\top}_{(AA^{-1})^\top} = I,$$

and this is true since we can invoke Lemma 2.19 to pull the transpositions out (below the curly braces) and then use that A and A^{-1} are inverse to each other, alongside with $I^\top = I$. □

3.3.2 The Inverse Theorem

We have already made the point that a 2×2 matrix is invertible if and only if its columns are linearly independent. This actually holds for $m \times m$ matrices in general. Even for $m = 1$, this makes sense. Saying that the columns of a 1×1 matrix $\begin{bmatrix} a \end{bmatrix}$ are linearly independent is a fancy way of saying that $a \neq 0$.

The following theorem can be thought of as characterizing "good" matrices (the ones for which Gauss elimination succeeds) in three different ways, connecting three important concepts. We already know from Theorem 3.5 that success of Gauss elimination is equivalent to statement (iii).

Theorem 3.11 (Inverse Theorem)**.** *Let A be an $m \times m$ matrix. The following statements are equivalent.*

(i) A is invertible.

(ii) For every $\mathbf{b} \in \mathbb{R}^m$, $A\mathbf{x} = \mathbf{b}$ has a unique solution $\mathbf{x}$.

(iii) The columns of A are linearly independent.

Proof. We establish equivalence through the following implications:

$$\begin{tikzcd}
\text{(i)} \arrow[r, Rightarrow] & \text{(ii)} \arrow[d, Rightarrow] \\
\text{(ii)} \arrow[u, Rightarrow] & \text{(iii)} \arrow[l, Rightarrow]
\end{tikzcd}$$

(i) ⇒ (ii): if A is invertible, the natural candidate for the unique solution of $A\mathbf{x} = \mathbf{b}$ is $\mathbf{x} = A^{-1}\mathbf{b}$. This is the m-dimensional analog of solving $ax = b$ via $x = b/a$. Indeed, this works. To show that $A^{-1}\mathbf{b}$ solves $A\mathbf{x} = \mathbf{b}$, we compute

$$A(A^{-1}\mathbf{b}) = (AA^{-1})\mathbf{b} = I\mathbf{b} = \mathbf{b}.$$

To prove uniqueness, take any solution $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{b}$. Multiplying by A^{-1} from both sides gives

$$\mathbf{x} = A^{-1}A\mathbf{x} = A^{-1}\mathbf{b}.$$

So there is no solution other than $A^{-1}\mathbf{b}$.

(ii) $\Rightarrow$ (iii): if $A\mathbf{x} = \mathbf{b}$ has a unique solution for every $\mathbf{b}$, this in particular holds for $\mathbf{b} = \mathbf{0}$. For this choice of $\mathbf{b}$, (ii) is saying that the columns of A are linearly independent, see Observation 3.2; so we have deduced (iii).

(iii) $\Rightarrow$ (ii): If the columns of A are linearly independent, Gauss elimination succeeds by Theorem 3.5, producing an equivalent system $U\mathbf{x} = \mathbf{c}$ with the upper triangular matrix U having nonzero diagonal entries. Back substitution as in Section 3.2.1 shows that there is a unique solution of $U\mathbf{x} = \mathbf{c}$ and hence also of $A\mathbf{x} = \mathbf{b}$.

(ii) $\Rightarrow$ (i): If $A\mathbf{x} = \mathbf{b}$ has a unique solution for all $\mathbf{b}$, we find vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m \in \mathbb{R}^m$ such that

$$A\mathbf{v}_1 = \underbrace{\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}}_{\mathbf{e}_1}, A\mathbf{v}_2 = \underbrace{\begin{bmatrix}0\\1\\\vdots\\0\end{bmatrix}}_{\mathbf{e}_2}, \ldots, A\mathbf{v}_m = \underbrace{\begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix}}_{\mathbf{e}_m} \quad \Rightarrow \quad A \underbrace{\begin{bmatrix}| & | & & |\\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m\\ | & | & & |\end{bmatrix}}_{B} = \underbrace{\begin{bmatrix}1 & 0 & \cdots & 0\\0 & 1 & \cdots & 0\\\vdots & \vdots & \ddots & \vdots\\0 & 0 & \cdots & 1\end{bmatrix}}_{I}.$$

So $AB = I$, and B is the obvious candidate for the inverse of A. We still need to show that $BA = I$ to conclude that $B = A^{-1}$ and that A is invertible according to Definition 3.7.

For this, we first compute $AI = IA = (AB)A = A(BA)$, so $A(I - BA) = 0$. Here, on top of associativity, we also use distributivity of matrix multiplication; see Lemma 2.22. Let the columns of $I - BA$ be $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m$. Then $A(I - BA) = 0$ reads as $A\mathbf{w}_j = \mathbf{0}$ for all j. The columns of A are linearly independent by (ii) $\Rightarrow$ (iii). Hence $\mathbf{w}_j = \mathbf{0}$ for all j, since $\mathbf{0}$ is the only solution of $A\mathbf{x} = \mathbf{0}$, see Observation 3.2. So all columns of $I - BA$ are $\mathbf{0}$, meaning that $BA = I$. □

In the previous proof, we have seen that $AB = I$ implies $BA = I$ **if** the columns of A are linearly independent. The next exercise shows that we do not need to require this condition. As you may realize in solving this exercise, $AB = I$ already implies that the columns of A are linearly independent.

Exercise 3.12. *Let A and B be two $m \times m$ matrices such that $AB = I$. Then we also have $BA = I$ and therefore $A^{-1} = B$ and $B^{-1} = A$ by Definition 3.7.*

3.4 LU and LUP decomposition

Here, we introduce two important decompositions of a square matrix with linearly independent columns, or of some version of it with exchanged rows: $A = LU$ if Gauss elimination succeeds without row exchanges, and $PA = LU$ if row exchanges are needed. Here, L is a lower triangular matrix, U is an upper triangular matrix, and P is a permutation matrix. This section can also be considered as a formal correctness proof of Gauss elimination.

Recall that Gauss elimination is trying to transform an $m \times m$ matrix A into an upper triangular matrix U, via row operations; see Section 3.2.2.

If this succeeds without row exchanges, we will be able to says precisely how A and U relate to each other. This will lead us to the LU decomposition. But even if there are row exchanges (the less nice case), or Gauss elimination fails altogether (the ugly case), we can still get something a bit weaker, namely an LUP decomposition.

Once we have an LU or LUP decomposition of A, we can solve $A\mathbf{x} = \mathbf{b}$ in time $O(m^2)$, for any given right-hand side $\mathbf{b}$. This is much faster then Gauss elimination which we have seen needs $O(m^3)$ time; see Section 3.2.4. We still need $O(m^3)$ time to compute the LU or LUP decomposition, but if we need to solve $A\mathbf{x} = \mathbf{b}$ for many different $\mathbf{b}$, this initial effort pays off quickly.

3.4.1 LU decomposition

Let's assume that Gauss elimination on $A\mathbf{x} = \mathbf{b}$ succeeds without row exchanges, transforming $A\mathbf{x} = \mathbf{b}$ into $U\mathbf{x} = \mathbf{c}$ with the same solutions, where U is an upper triangular matrix.

More concretely, we have obtained U as a product of some elimination matrices and A; in the example of Section 3.2.2, this product looks as follows:

$$\underbrace{\begin{bmatrix}1&0&0\\0&1&0\\0&-1&1\end{bmatrix}}_{\substack{\text{subtract } 1\cdot\text{(row 2)}\\ \text{from (row 3)}}}\;\underbrace{\begin{bmatrix}1&0&0\\0&1&0\\-1&0&1\end{bmatrix}}_{\substack{\text{subtract } 1\cdot\text{(row 1)}\\ \text{from (row 3)}}}\;\underbrace{\begin{bmatrix}1&0&0\\-2&1&0\\0&0&1\end{bmatrix}}_{\substack{\text{subtract } 2\cdot\text{(row 1)}\\ \text{from (row 2)}}}\;\underbrace{\begin{bmatrix}2&3&4\\4&11&14\\2&8&17\end{bmatrix}}_{A} = \underbrace{\begin{bmatrix}2&3&4\\0&5&6\\0&0&7\end{bmatrix}}_{U}.$$

For another 3×3 matrix A, the multiples that we subtract will be different, but the general pattern is the same:

$$\underbrace{\begin{bmatrix}1&0&0\\0&1&0\\0&-c_{32}&1\end{bmatrix}}_{\substack{\text{subtract } c_{32}\cdot\text{(row 2)}\\ \text{from (row 3)}}}\;\underbrace{\begin{bmatrix}1&0&0\\0&1&0\\-c_{31}&0&1\end{bmatrix}}_{\substack{\text{subtract } c_{31}\cdot\text{(row 1)}\\ \text{from (row 3)}}}\;\underbrace{\begin{bmatrix}1&0&0\\-c_{21}&1&0\\0&0&1\end{bmatrix}}_{\substack{\text{subtract } c_{21}\cdot\text{(row 1)}\\ \text{from (row 2)}}}\;A = U.$$

Multiplying the three elimination matrices with each other yields

$$\begin{bmatrix} 1 & 0 & 0 \\ -c_{21} & 1 & 0 \\ c_{32}c_{21} - c_{31} & -c_{32} & 1 \end{bmatrix} A = U. \tag{3.4}$$

The "complicated" matrix on the left is the one resulting from applying the second and the third elimination step to the elimination matrix of the first step (and this is probably the easiest way to compute the matrix). But here comes the magic: The inverse of the complicated matrix is a very simple matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ -c_{21} & 1 & 0 \\ c_{32}c_{21} - c_{31} & -c_{32} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ c_{21} & 1 & 0 \\ c_{31} & c_{32} & 1 \end{bmatrix}.$$

To verify this, you can multiply the two matrices and check that the result is indeed the identity matrix. Hence, multiplying both sides of (3.4) with the simple inverse yields

$$A = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ c_{21} & 1 & 0 \\ c_{31} & c_{32} & 1 \end{bmatrix}}_{L} U.$$

This means, we have obtained a decomposition of A in the form $A = LU$. The matrix U is the upper triangular one resulting from Gauss elimination, and L is a lower triangular matrix with 1's on the diagonal; below the diagonal, L records the multiples that we have used for the row subtractions throughout the elimination steps.

The question is whether this always works; maybe it does for $m = 3$ but not in general. To show that it works in general, we need a theorem with a proof. If we are only interested in A, we can simply remove lines 12 and 22 (affecting b) from the elimination procedure in Table 3.6. This gives us a version of Gauss elimination that transforms only A instead of A and b, and this is the one we refer to in the next theorem.

Theorem 3.13 (LU decomposition). *Let A be an $m \times m$ matrix on which Gauss elimination as in Table 3.6 succeeds without row exchanges, resulting in an upper triangular matrix U. Let c_{ij} (computed in Line 18) be the multiple of row j that we subtract from row $i > j$ when we eliminate in column j. Then $A = LU$ where*

$$L = \begin{bmatrix} 1 & & & \\ c_{21} & 1 & & \\ \vdots & & \ddots & \\ c_{m1} & \cdots & c_{m,m-1} & 1 \end{bmatrix}.$$

More formally, $L = [\ell_{ij}]_{i=1,j=1}^{m\ \ m}$ where

$$\ell_{ij} = \begin{cases} 0 & \text{if } i < j \\ 1 & \text{if } i = j \\ c_{ij} & \text{if } i > j \end{cases}.$$

Proof. We look at a fixed row i. Whenever we change row i during Gauss elimination, we subtract $c_{ij} \cdot$ (row j) from it, for some previous row j. At this point, row j has already been "finalized", meaning that it is the j-th row of our resulting matrix U. Pictorially, the situation is this, with $u_{jj} \neq 0$ being the current pivot:

$$\begin{array}{r|cccccc|l}
 & u_{11} & \cdots & & & & & \leftarrow \text{finalized (in } U) \\
 & 0 & u_{22} & \cdots & & & & \leftarrow \text{finalized (in } U) \\
 & 0 & 0 & \ddots & & & & \vdots \\
\text{row } j & 0 & 0 & \cdots & \mathbf{u_{jj}} & \cdots & u_{jm} & \leftarrow \text{finalized (in } U) \\
\vdots & & & & & & & \\
\text{row } i & 0 & 0 & \cdots & \star_{ij} & \cdots & \star_{im} & \leftarrow \text{now subtract } c_{ij} \cdot (\text{row } j)
\end{array}$$

So if we track what happens to row i (initially in A), we see multiples of finalized rows $1, 2, \ldots, i-1$ being subtracted, after which we end up with row i in U:

$$\begin{array}{crcll}
 & & & (\text{row } i) \text{ in } A & \text{initially} \\
- & c_{i1} & \cdot & (\text{row } 1) \text{ in } U & \text{step } 1 \\
- & c_{i2} & \cdot & (\text{row } 2) \text{ in } U & \text{step } 2 \\
\vdots & & & & \\
- & c_{i,i-1} & \cdot & (\text{row } i-1) \text{ in } U & \text{step } i-1 \\
= & & & (\text{row } i) \text{ in } U & \text{finalized.}
\end{array}$$

If we solve this equation for the first term, (row i) in A, we see that this row is a linear combination of the first i rows of U. In matrix notation, the linear combination can be written as follows:

$$(\text{row } i) \text{ of } A = \underbrace{\begin{bmatrix} c_{i1} & c_{i2} & \cdots & c_{i,i-1} & 1 & 0 & \cdots & 0 \end{bmatrix}}_{\text{row vector}} U.$$

Doing this for all rows of A and combining the m row vector equations into one matrix equation gives the desired result:

$$A = \underbrace{\begin{bmatrix} 1 & & & \\ c_{21} & 1 & & \\ \vdots & & \ddots & \\ c_{m1} & \cdots & c_{m,m-1} & 1 \end{bmatrix}}_{L} U.$$

□

In other words, Gauss elimination (without row exchanges) actually computes an LU decomposition of A while transforming A to U. This all happens in time $O(m^3)$, see Theorem 3.6.

Solving $A\mathbf{x} = \mathbf{b}$ from $A = LU$. Once we have an LU decomposition of A, we can use it to solve $A\mathbf{x} = \mathbf{b}$ for *any* given right-hand side $\mathbf{b}$ in $O(m^2)$ time which is much faster than Gauss elimination. This is very useful if we need to solve many systems with the same matrix A.

To see this, we write $A\mathbf{x} = \mathbf{b}$ as

$$L\,\underbrace{U\mathbf{x}}_{\mathbf{y}} = \mathbf{b}.$$

We first solve $L\mathbf{y} = \mathbf{b}$ for $\mathbf{y}$. Since L is lower triangular, we can do this using *forward substitution*. This works exactly like back substitution (Section 3.2.1), except that it finds the solution in forward order $y_1, y_2, \ldots$, because L is lower instead of upper triangular. Since L has 1's on the diagonal, this succeeds, and we don't even need divisions.

Once we have $\mathbf{y}$, we now solve $U\mathbf{x} = \mathbf{y}$ using back substitution to obtain $\mathbf{x}$. Forward and back substitution only require (m^2) arithmetic steps each and therefore run in $O(m^2)$ time; see Section 3.2.4. One way to think about solving $L\mathbf{y} = \mathbf{b}$ is that we simply *replay* the row subtractions that Gauss elimination previously did on A, except that we now do them on $\mathbf{b}$ as well (putting back lines 12 and 22), leading to $\mathbf{y} = \mathbf{c}$ in the end.

Beyond LU. What happens if Gauss elimination succeeds but needs some row exchanges on the way? The replay approach still works and solves $A\mathbf{x} = \mathbf{b}$ in time $O(m^2)$ if we remember all row subtractions *and* row exchanges that happened on the way. With $A = LU$, the matrix L was our "memory", but if there are row exchanges, this no longer works.

The proof of Theorem 3.13 breaks down, since it can only deal with elimination matrices, but not with permutation matrices in between them. There is in general no way to fix this: there are matrices for which Gauss elimination succeeds (with row exchanges), but there is no LU decomposition.

As an example, consider a matrix of the the form

$$A = \begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix},$$

where a is an arbitrary number. Gauss elimination is done after one row exchange.

Let's try to write A in the form $A = LU$ where L is lower triangular and U is upper triangular:

$$\underbrace{\begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \ell_{11} & 0 \\ \ell_{21} & \ell_{22} \end{bmatrix}}_{L}\underbrace{\begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}}_{U}.$$

These are 4 equations, one for each entry of A, and we consider three of them:

$$\begin{aligned} &0 = \ell_{11}u_{11}, \quad 1 = \ell_{11}u_{12}, \\ &1 = \ell_{21}u_{11}. \end{aligned}$$

The equation $0 = \ell_{11}u_{11}$ can only hold if at least one of ℓ_{11} and u_{11} is 0, but then at least one of the two other equations fails. So there is no way to get $A = LU$ in this case.

But we can always get an LU decomposition after exchanging some rows of A first, and this will give us the LUP decomposition

$$PA = LU,$$

where P is a permutation matrix. This is the "memory" of all that happened when there were also row exchanges. The permutation matrix P will record the row exchanges, and L records the row subtractions.

Before we can formally derive the LUP decomposition, we need to introduce permutation matrices.

3.4.2 Permutations and permutation matrices

Previously, we have only used permutation matrices that exchange two rows, but now we need to deal with general permutation matrices. We start with permutations.

Definition 3.14 (Permutation)**.** *A permutation of* $[m] = \{1, 2, \ldots, m\}$ *is a bijective function* $\pi : [m] \to [m]$.

Bijective means that the function values cover all of $[m]$: no value is missing, and no value appears twice. Here is an example for $m = 5$.

i	1	2	3	4	5
$\pi(i)$	2	3	5	4	1

(3.5)

Hence, a permutation can be thought of as reordering the sequence $1, 2, \ldots, m$ into $\pi(1), \pi(2), \ldots, \pi(m)$. In the example, the reordered sequence is $2, 3, 5, 4, 1$.

There are

$$m! = 1 \cdot 2 \cdots m$$

permutations of $[m]$. To see this, we can for example argue as follows: there are m ways of putting 1 into the sequence $\pi(1), \pi(2), \ldots, \pi(m)$. In the previous example, 1 was put last, but we could have put it at any of the 5 positions in the lower row of the table.

For each of the m ways of putting 1, there are $m - 1$ ways of putting 2, so there are $m(m-1)$ ways of putting 1 *and* 2. For each of these ways, there are $m - 2$ ways of putting 3, and so on. In the end, the number of ways of putting all m numbers is $m!$.

If π, π' are two permutations of $[m]$, then their *composition* $\pi' \circ \pi$ is again a permutation. The composition is defined as

$$(\pi' \circ \pi)(i) = \pi'(\pi(i)),$$

the function that first applies π and then π'. To show that this is a permutation, we need to check the bijection: for every k, there is exactly one i with $k = \pi'(\pi(i))$. Indeed, there is exactly one j with $k = \pi'(j)$, so we must have $\pi(i) = j$; again, there is exactly one i doing that.

Definition 3.15 (Permutation matrix). *Let* $\pi : [m] \to [m]$ *be a permutation. The* permutation matrix *associated with* π *is the* $m \times m$ *matrix* $P = [p_{ij}]_{i=1,j=1}^{m\ \ m}$ *with*

$$p_{ij} = \begin{cases} 1 & \textit{if } j = \pi(i) \\ 0 & \textit{otherwise} \end{cases} .$$

Vice versa, we also say that π *is associated with* P*.*

For example,

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

is the permutation matrix associated with π in (3.5). The row/column pairs (i, j) where we find 1's are $(1, 2), (2, 3), (3, 5), (4, 4), (5, 1)$—exactly the pairs (i, j) where $j = \pi(i)$.

Since a permutation is bijective, a permutation matrix has a single 1 in each row and column. The 1 in row i is in column $\pi(i)$, and the 1 in column j is in the unique row i such that $\pi(i) = j$. Vice versa, every matrix with a single 1 in each row and column is a permutation matrix. The associated permutation is given by the positions of the 1's in each row.

In this sense, a permutation matrix can be considered as the graph of its associated permutation, with inputs on the "row axis" and outputs on the "column axis", se Figure 3.2.

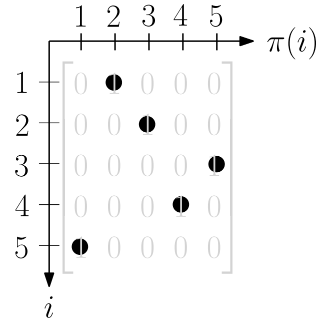


Figure 3.2: A permutation matrix can be interpreted as the graph of its associated permutation: The 1 in row i indicates the function value $\pi(i)$, visualized with a black dot in the graph.

The linear transformation T_P of a permutation matrix P (Definition 2.25) reorders the input vector according to the associated permutation. For example,

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} x_2 \\ x_3 \\ x_5 \\ x_4 \\ x_1 \end{bmatrix}}_{T_P(\mathbf{x})}.$$

It follows that if P is an $m \times m$ permutation matrix and A any $m \times n$ matrix, then PA is the matrix arising from A by reordering the rows according to the associated permutation.

In summary, a permutation matrix is the "linear algebra way" of looking at a permutation, but it contains the same information. In particular, there are also $m!$ permutation matrices of size $m \times m$. The identity matrix I is a special permutation matrix; its associated permutation is the identity permutation with $\pi(i) = i$ for all i.

What about P^{-1}, the inverse of the permutation matrix P associated with π? If it exists, then applying P^{-1} to $P\mathbf{x}$ (the input vector after reordering) must give $P^{-1}P\mathbf{x} = I\mathbf{x} = \mathbf{x}$ (the input vector before reordering), so P^{-1} is undoing the reordering. Such a matrix P^{-1} therefore indeed exists. It is again a permutation matrix, associated with π^{-1}, the inverse of π: If $\pi(i) = j$, then $\pi^{-1}(j) = i$. Table 3.7 shows how this looks like for the example (3.5):

i	1	2	3	4	5
$\pi(i)$	2	3	5	4	1

j	1	2	3	4	5
$\pi^{-1}(j)$	5	1	2	4	3

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad P^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Table 3.7: The inverse permutation matrix corresponds to the inverse permutation. While $\pi(5) = 1$ (last column of π's value table), we have $\pi^{-1}(1) = 5$ (first column of π^{-1}'s value table).

In this example, P^{-1} is the transpose of P, and this is true in general.

Lemma 3.16. *Let P be a permutation matrix. Then $P^{-1} = P^\top$.*

Proof. We check that $PP^\top = I$ ($P^\top P = I$ follows by a similar argument, or directly from Exercise 3.12). What we need to check for this is

$$(i\text{-th row of } P) \cdot (j\text{-th column of } P^\top) = \delta_{ij} = \left\{ \begin{array}{ll} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{array} \right. .$$

By Definition 2.11 of the transpose, the j-th column of $P^\top$ is the j-th row of P, hence, we need to check that

$$(i\text{-th row of } P) \cdot (j\text{-th row of } P) = \delta_{ij}.$$

This is now easy: row i of P is the standard unit vector $\mathbf{e}_{\pi(i)}$, where π is the associated permutation. Thus, $\mathbf{e}_{\pi(i)} \cdot \mathbf{e}_{\pi(i)} = 1$ for all i. For $i \neq j$, $\mathbf{e}_{\pi(i)} \cdot \mathbf{e}_{\pi(j)} = 0$ since $\pi(i) \neq \pi(j)$ (π is a bijection). □

Whenever we have a square matrix

$$M = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \\ | & | & & | \end{bmatrix}$$

such that

$$\mathbf{v}_i \cdot \mathbf{v}_j = \delta_{ij},$$

we can argue in the same way that $M^{-1} = M^\top$. Such matrices are called *orthogonal matrices* and will play an important role in the second part of the course. Here, we have seen that permutation matrices form a (simple) class of orthogonal matrices.

We conclude the treatment of permutation matrices with another simple but important lemma.

Lemma 3.17. *Let P, P' be $m \times m$ permutation matrices with associated permutations π, π'. Then PP' is a permutation matrix as well, associated with the permutation $\pi' \circ \pi$.*

Before we prove this, let us clear a potential confusion: You may have expected the associated permutation to be $\pi \circ \pi'$, and this would indeed be the result if we had decided to read a permutation matrix *column-wise* instead of row-wise, in order to obtain its associated permutation. But this decision would have had other counter-intuitive effects.

At the bottom of this issue is the following: there are two natural ways of describing an ordering of $[m]$ by a permutation π. Let's look at the ordering

$$2, 3, 5, 4, 1$$

that we have considered before. We have chosen π such that the i-th element in the ordering is $\pi(i)$. This gives

$$\begin{array}{c|c|c|c|c|c} i & 1 & 2 & 3 & 4 & 5 \\ \hline \pi(i) & 2 & 3 & 5 & 4 & 1 \end{array}.$$

Alternatively, we could have chosen π such that element i occurs at position $\pi(i)$ in the ordering. This gives

$$\begin{array}{c|c|c|c|c|c} i & 1 & 2 & 3 & 4 & 5 \\ \hline \pi(i) & 5 & 1 & 2 & 4 & 3 \end{array},$$

exactly the inverse of the previous permutation.

The upshot is that if someone just says "the permutation $2, 3, 5, 4, 1$", then we should clarify what they mean.

In any case, the important consequence of Lemma 3.17 is that the product of two permutation matrices is again a permutation matrix. Let's prove it.

Proof. We have

$$PP' \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = P \underbrace{\begin{bmatrix} x_{\pi'(1)} \\ x_{\pi'(2)} \\ \vdots \\ x_{\pi'(m)} \end{bmatrix}}_{\mathbf{y}} = P \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} y_{\pi(1)} \\ y_{\pi(2)} \\ \vdots \\ y_{\pi(m)} \end{bmatrix} = \begin{bmatrix} x_{\pi'(\pi(1))} \\ x_{\pi'(\pi(2))} \\ \vdots \\ x_{\pi'(\pi(m))} \end{bmatrix} = \begin{bmatrix} x_{(\pi'\circ\pi)(1)} \\ x_{(\pi'\circ\pi)(2)} \\ \vdots \\ x_{(\pi'\circ\pi)(m)} \end{bmatrix}.$$

In the second-to-last equality, we use the definition of $\mathbf{y}$: $y_i = x_{\pi'(i)}$ for all i. □

3.4.3 LUP decomposition

Now we are prepared to establish the "memory" of Gauss elimination in the general case.

Theorem 3.18 (LUP decomposition)**.** *Let A be an $m \times m$ matrix with linearly independent columns, $m \geq 1$. There exist three $m \times m$ matrices P, L, U such that*

$$PA = LU,$$

where P is a permutation matrix, L a lower triangular matrix with 1's on the diagonal, and U an upper triangular matrix with nonzero diagonal entries.

The idea is the following: running Gauss elimination yields

$$U = E_{m-1}P_{m-1}E_{m-2}P_{m-2} \cdots E_1P_1A,$$

where P_j is the permutation matrix for the row exchange in column j ($P_j = I$ if there is no row exchange), and E_j is the product of all elimination matrices used to produce the zeros below the diagonal in column j.

What if we move all row exchanges to the very beginning and then do Gauss elimination on the matrix PA where $P = P_{m-1}P_{m-2} \cdots P_1$, the permutation matrix that summarizes all the row exchanges? Our hope is that this now works without any further row exchanges, in which case Theorem 3.13 gives us an LU decomposition of PA.

Now we rigorously show that our hope is reality. The following proof is from the textbook *Introduction to Algorithms* by Cormen, Leiserson, Rivest, and Stein [CLRS22, Section 28.1]. Their proof works for *any* square matrix A, resulting in a matrix U that may have some 0's on the diagonal. To stay consistent with our chapter's philosophy, we present the proof only for the case where A is invertible and Gauss elimination succeeds; this is the case covered by Theorem 3.18.

Proof. We proceed by induction on m. For the base case $m = 1$, we have $A = \begin{bmatrix} a \end{bmatrix}$ for some number $a \neq 0$, so we can choose $P = L = \begin{bmatrix} 1 \end{bmatrix}$ (the 1×1 identity matrix) and $U = A$ to satisfy $PA = LU$ as required.

If $m > 1$, we start with a row exchange to transform A into

$$P_1 A = \left[\begin{array}{c|c} a & -\,\mathbf{u}\,- \\ \hline \vert & \\ \mathbf{v} & B \\ \vert & \end{array}\right],$$

with $a \neq 0$. P_1 is the permutation matrix that does the row exchange (we have $P_1 = I$ if no row exchange is necessary). We know that such a row exchange is possible: Gauss elimination succeeds whenever A has linearly independent columns (Theorem 3.5).

Next, we apply elimination in the first column to turn $\mathbf{v}$ into $\mathbf{0}$, via

$$\underbrace{\left[\begin{array}{c|c} 1 & -\,\mathbf{0}\,- \\ \hline \vert & \\ -\frac{1}{a}\mathbf{v} & I \\ \vert & \end{array}\right]}_{E_1} P_1 A = \left[\begin{array}{c|c} a & -\,\mathbf{u}\,- \\ \hline \vert & \\ \mathbf{0} & A' \\ \vert & \end{array}\right]. \tag{3.6}$$

The matrix E_1 is collecting all $m - 1$ elimination matrices we need for that in one matrix. The effect of premultiplication with E_1 is that $(\frac{1}{a}v_i) \cdot$ (row 1) is subtracted from row $i + 1$, for $i = 1, \ldots, m - 1$. As needed, this cancels all the v_i's below a.

Since A' is an $(m-1) \times (m-1)$ matrix with linearly independent columns (this follows from Corollary 3.4; try to do the argument!), we can now use the induction hypothesis to find $(m-1) \times (m-1)$ matrices P', L', U' such that $P'A' = L'U'$, where P' is a permutation matrix, L' a lower triangular matrix with 1's on the diagonal, and U' an upper triangular matrix with nonzero diagonal entries. Applying the extended permutation matrix

$$P'_+ = \left[\begin{array}{c|c} 1 & -\,\mathbf{0}\,- \\ \hline \vert & \\ \mathbf{0} & P' \\ \vert & \end{array}\right]$$

to both sides of (3.6), we therefore get

$$P'_+ E_1 P_1 A = \left[\begin{array}{c|c} 1 & -\,\mathbf{0}\,- \\ \hline \vert & \\ \mathbf{0} & P' \\ \vert & \end{array}\right] \left[\begin{array}{c|c} a & -\,\mathbf{u}\,- \\ \hline \vert & \\ \mathbf{0} & A' \\ \vert & \end{array}\right] = \left[\begin{array}{c|c} a & -\,\mathbf{u}\,- \\ \hline \vert & \\ \mathbf{0} & P'A' \\ \vert & \end{array}\right] = \left[\begin{array}{c|c} a & -\,\mathbf{u}\,- \\ \hline \vert & \\ \mathbf{0} & L'U' \\ \vert & \end{array}\right]. \tag{3.7}$$

So far, nothing much has actually happened (but verify the first equality for yourself!). One way we can think about it is this: The induction hypothesis has already moved the row exchanges in steps $2, \ldots, m-1$ of Gauss elimination to the beginning of step 2. Since we simply assume the induction hypothesis, this has happened "for free" and transformed

$$U = E_{m-1}P_{m-1}E_{m-2}P_{m-2}\cdots E_1P_1A$$

into

$$U = E'_{m-1}E'_{m-2}\cdots E'_2\underbrace{P_{m-1}P_{m-2}\cdots P_2}_{P'_+}E_1P_1A.$$

There is just one last step missing, but that one we have to do ourselves: get E_1 past P'_+.

Claim: With $\mathbf{w} = -\frac{1}{a}\mathbf{v}$ and $\mathbf{w}' = P'(-\frac{1}{a}\mathbf{v})$ the result of applying P' to $\mathbf{w}$, we have

$$\underbrace{\left[\begin{array}{c|c} 1 & \text{---}\,\mathbf{0}\,\text{---} \\ \hline | & \\ \mathbf{0} & P' \\ | & \end{array}\right]}_{P'_+}\underbrace{\left[\begin{array}{c|c} 1 & \text{---}\,\mathbf{0}\,\text{---} \\ \hline | & \\ -\mathbf{w} & I \\ | & \end{array}\right]}_{E_1} = \underbrace{\left[\begin{array}{c|c} 1 & \text{---}\,\mathbf{0}\,\text{---} \\ \hline | & \\ -\mathbf{w}' & I \\ | & \end{array}\right]}_{E'_1}\underbrace{\left[\begin{array}{c|c} 1 & \text{---}\,\mathbf{0}\,\text{---} \\ \hline | & \\ \mathbf{0} & P' \\ | & \end{array}\right]}_{P'_+}.$$

A *claim* is a "sub-lemma" within a proof that doesn't deserve full lemma status, since it is needed only locally. A claim is also proved locally. To do this here, we compute both products. P'_+E_1 results from E_1 by permuting the rows, leaving the first row in place and permuting the other ones using P'. This yields

$$P'_+E_1 = \left[\begin{array}{c|c} 1 & \text{---}\,\mathbf{0}\,\text{---} \\ \hline | & \\ -\mathbf{w}' & P' \\ | & \end{array}\right].$$

$E'_1P'_+$, on the other hand, results from P'_+ by subtracting $w'_i \cdot$ (row 1) from row $i+1$, for $i = 1, 2, \ldots, m-1$. As row 1 is $\left[\begin{array}{c|c} 1 & \text{---}\,\mathbf{0}\,\text{---}\end{array}\right]$, this only affects the first column and replaces $\mathbf{0}$ by $-\mathbf{w}'$ there, hence

$$E'_1P'_+ = \left[\begin{array}{c|c} 1 & \text{---}\,\mathbf{0}\,\text{---} \\ \hline | & \\ -\mathbf{w}' & P' \\ | & \end{array}\right]$$

as well.

Applying the claim to (3.7) gives

$$E'_1\underbrace{P'_+P_1}_{P}A = \left[\begin{array}{c|c} a & \text{---}\,\mathbf{u}\,\text{---} \\ \hline | & \\ \mathbf{0} & L'U'. \\ | & \end{array}\right]. \qquad (3.8)$$

We have already found the desired permutation matrix $P = P'_+P_1$, using that a product of two permutation matrices is a permutation matrix (Lemma 3.17). It remains to find L and U. For this, we move E'_1 to the other side, by multiplying both sides with

$$(E'_1)^{-1} = \left[\begin{array}{c|c} 1 & -\mathbf{0}- \\ \hline | & \\ \mathbf{w}' & I \\ | & \end{array}\right].$$

This formula for the inverse is not surprising: to undo *subtractions* of multiples of the first row from the rows below, we simply *add* the same multiples back.

We can also easily verify that the right-hand side of (3.8) can be decomposed as

$$\left[\begin{array}{c|c} a & -\mathbf{u}- \\ \hline | & \\ \mathbf{0} & L'U'. \\ | & \end{array}\right] = \left[\begin{array}{c|c} 1 & -\mathbf{0}- \\ \hline | & \\ \mathbf{0} & L' \\ | & \end{array}\right] \left[\begin{array}{c|c} a & -\mathbf{u}- \\ \hline | & \\ \mathbf{0} & U' \\ | & \end{array}\right].$$

After this, (3.8) becomes

$$PA = \underbrace{(E'_1)^{-1} \left[\begin{array}{c|c} 1 & -\mathbf{0}- \\ \hline | & \\ \mathbf{0} & L' \\ | & \end{array}\right]}_{\text{computation as for } E'_1P'_+ \text{ in Claim}} \left[\begin{array}{c|c} a & -\mathbf{u}- \\ \hline | & \\ \mathbf{0} & U' \\ | & \end{array}\right] = \underbrace{\left[\begin{array}{c|c} 1 & -\mathbf{0}- \\ \hline | & \\ \mathbf{w}' & L' \\ | & \end{array}\right]}_{L} \underbrace{\left[\begin{array}{c|c} a & -\mathbf{u}- \\ \hline | & \\ \mathbf{0} & U' \\ | & \end{array}\right]}_{U}.$$

Hence we have also found L and U as desired. □

Admittedly, this proof was somewhat technical. But the alternative (believing it to be obvious that all row exchanges can be moved to the beginning) is not convincing.

Solving $A\mathbf{x} = \mathbf{b}$ from $PA = LU$. This is very similar to what we did in Section 3.4.1 from $A = LU$, except that it now works for every matrix A on which Gauss elimination succeeds.

We multiply with $P^{-1} = P^\top$ (see Lemma 3.17 for this formula) to obtain $A = P^\top LU$, hence we can write $A\mathbf{x} = \mathbf{b}$ as

$$P^\top \underbrace{L\,\underbrace{U\mathbf{x}}_{\mathbf{y}}}_{\mathbf{z}} = \mathbf{b}.$$

We first solve $P^\top\mathbf{z} = \mathbf{b}$ for $\mathbf{z}$. In fact, there's nothing to solve: $\mathbf{z} = P\mathbf{b}$, so we permute $\mathbf{b}$ in the same way we have permuted the rows of A. After this, we proceed as before: we solve $L\mathbf{y} = \mathbf{z}$ for $\mathbf{y}$ using forward substitution and then solve $U\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$, with back substitution. All this can be done in $O(m^2)$ time, so again much faster than solving $A\mathbf{x} = \mathbf{b}$ from scratch using Gauss elimination.

3.5 Gauss-Jordan elimination

In this section, we present an algorithm for solving any system $A\mathbf{x} = \mathbf{b}$ (or detecting that there is no solution). The algorithm is very similar in spirit to Gauss elimination. The most important theoretical contribution of Gauss-Jordan elimination is a reduction of A to a unique standard form from which we can easily read off many properties of A that are difficult to see directly. On top of providing an efficient way of solving $A\mathbf{x} = \mathbf{b}$, this standard form also yields the CR decomposition of A.

So far, we have seen how to solve systems $A\mathbf{x} = \mathbf{b}$ where A is a square matrix with linearly independent columns. This is a very nice case in the sense that there is always a unique solution $\mathbf{x}$ that can be found with Gauss elimination; see Theorem 3.5 and the Inverse Theorem 3.11.

We will now see an algorithm for the general case (A may be non-square and/or may have linearly dependent columns). This is a simple extension of Gauss elimination.

3.5.1 (Reduced) row echelon form

In Gauss elimination, we are trying to transform a square matrix A into an upper triangular matrix U with nonzero diagonal entries, using row operations repeatedly. This fails if A has linearly dependent columns. Now, we want to define a "standard form" into which we can transform *every* matrix, even if it has linearly dependent columns, or is not a square matrix. This standard form is the *row echelon form* (REF).

Figure 3.3 gives an example of a matrix in REF. This will make it easier to understand the subsequent definition.

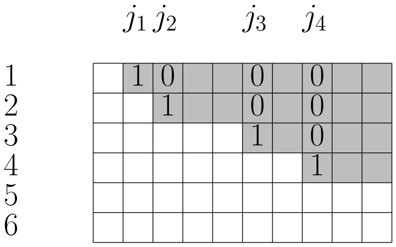


Figure 3.3: A 6×10 matrix in row echelon form $\mathrm{REF}(2, 3, 6, 8)$. White entries are 0, unlabeled gray entries can be any numbers. Removing the last two rows yields a 4×10 matrix in $\mathrm{RREF}(2, 3, 6, 8)$.

Definition 3.19 (Row echelon and reduced row echelon form). *Let $R = [r_{ij}]_{i=1,j=1}^{m\ \ n}$ be an $m \times n$ matrix. R is in* row echelon form *(*REF*) if the following holds: There exist $r \leq m$ column indices $1 \leq j_1 < j_2 < \cdots < j_r \leq n$ such that the following two statements hold:*

(i) For $i = 1, 2, \ldots, r$, we have $r_{ij_i} = 1$.

(ii) For all i, j, we have $r_{ij} = 0$ whenever $i > r$ or $j < j_i$ or $j = j_k$ for some $k > i$.

If $r = m$, R is in reduced row echelon form *(*RREF*). If we want to describe the shape of R precisely, we say that R is in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$ or $\mathrm{RREF}(j_1, j_2, \ldots, j_m)$.*

It's easiest to understand REF by looking at the matrix row by row. Row i only has 0s if $i > r$. This case happens for rows 5 and 6 in Figure 3.3.

Otherwise, row i starts with $j_i - 1$ 0s. In column j_i, we then have a 1 (first gray entry in the row). After this, we can have any entries, unless we are in some column j_k: there we again need to have a 0. You can check that the matrix in Figure 3.3 has this behavior for all rows.

As a consequence, column j_i equals $\mathbf{e}_i$, the i-th standard unit vector. The shape of a matrix in REF resembles that of an upper triangular matrix in the sense that there are prescribed 0s to the lower left (the white area).

Removing rows 5 and 6 in Figure 3.3 yields a matrix in RREF. Generally, RREF does not allow any zero rows.

The $m \times m$ identity matrix I is in $\mathrm{RREF}(1, 2, \ldots, m)$, while the $m \times n$ zero matrix is in $\mathrm{REF}()$, meaning that $r = 0$.

Matrices in REF and RREF are very nice in the sense that we can read off many properties easily. Here is a first example.

Observation 3.20. *A matrix R in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$ has rank r.*

Proof. Recall that the rank of a matrix is the number of independent columns, the ones that are not linear combinations of previous columns (Definition 2.9). In R, the independent columns are precisely the ones with indices $j_1, j_2, \ldots, j_r$. Indeed, in each of these columns, R makes a "downward step" (has a nonzero entry where all previous columns have zeros). Such a downward step column is therefore not a linear combination of the previous columns. But any other column is immediately seen to be a linear combination of columns $j_1, j_2, \ldots, j_r$, since these are simply the first r standard unit vectors; their linear combinations are all vectors in $\mathbb{R}^m$ with 0-entries at coordinates $r + 1, r + 2, \ldots, m$. □

3.5.2 Direct solution

It is easy to solve $A\mathbf{x} = \mathbf{b}$ whenever A is in REF. Much easier even than back substitution for an upper triangular matrix.

Suppose that A is an $m \times n$ matrix in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$ There are two cases: if $b_i \neq 0$ for some $i > r$, there is no solution. This is because the i-th row of A is zero for all $i > r$, so the i-th entry of $A\mathbf{x}$ is zero for $i > r$, no matter what $\mathbf{x}$ is.

If $b_i = 0$ for all $i > r$, there is a solution: we define $\mathbf{x}$ by

$$x_j = \begin{cases} b_i, & \text{if } j = j_i \\ 0, & \text{otherwise.} \end{cases}$$

See Figure 3.4 for an illustration.

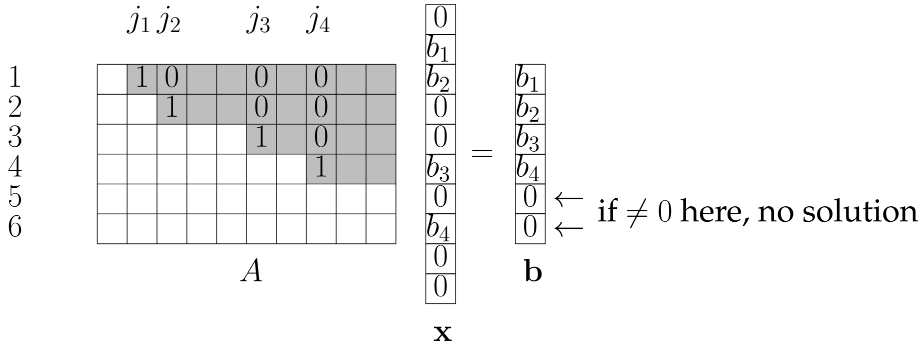


Figure 3.4: Direct solution of $A\mathbf{x} = \mathbf{b}$ when A is in RREF

.

This works: since the j_i-th column of A is $\mathbf{e}_i$, this yields

$$A\mathbf{x} = \sum_{i=1}^{r} b_i \mathbf{e}_i = \sum_{i=1}^{m} b_i \mathbf{e}_i = \mathbf{b}.$$

For the middle equality, we use that $b_{r+1}, b_{r+2}, \ldots, b_m = 0$.

The vector $\mathbf{x}$ just constructed may not be the only solution, but it is the *canonical* one.

3.5.3 Elimination

Here, we proceed similar to Gauss elimination and use row operations to transform any system $A\mathbf{x} = \mathbf{b}$ into an equivalent system $R_0\mathbf{x} = \mathbf{c}$ where R_0 is in REF. After this, we can use direct solution (see previous section) to either report that the system is unsolvable, or to compute the canonical solution. The resulting matrix is called R_0, since it may have zero rows at the end. We reserve the name R for the matrix obtained after removing the zero row.

For Gauss elimination, we have first shown examples and then computer code. Finally, Theorem 3.18 can be considered as the "official" correctness proof of Gauss elimination. Here, we again start with an example but skip the computer code. It is very similar to the one for Gauss elimination but more lengthy. After the example, we therefore proceed directly to the correctness proof. The good thing is that this one is simpler than the one for Gauss elimination and at the same time closer to the actual algorithm.

We only show how elimination transforms A into R_0. To transform $\mathbf{b}$ into $\mathbf{c}$, we apply the same row operations. One elegant way of doing this is to directly work with the extended matrix $[A|\mathbf{b}]$ that has $\mathbf{b}$ as an extra column. But even after transforming only A (and computing the product of all row operation matrices on the way), we will be able to efficiently solve $A\mathbf{x} = \mathbf{b}$ for every $\mathbf{b}$; see Theorem 3.23 below. This is similar to the case of Gauss elimination where the resulting LU or LUP decomposition of A provides a way of solving $A\mathbf{x} = \mathbf{b}$ for every $\mathbf{b}$; see Sections 3.4.1 and 3.4.3.

As in Gauss elimination, we proceed column by column. The nonzero pivots that we find will determine the "downward step" columns $j_1, j_2, \ldots$ of the resulting matrix in REF. We maintain a number r, initially equal to 0, counting how many downwards steps we have already made. The next downward step will then be in some upcoming column of row $r+1$.

Our example matrix is the 3×5 matrix

$$A = \begin{bmatrix} 2 & 4 & 2 & 2 & -2 \\ 6 & 12 & 6 & 7 & 1 \\ 4 & 8 & 2 & 2 & 6 \end{bmatrix}. \tag{3.9}$$

In column 1, we find a nonzero pivot in row 1 that we will use for elimination in this column. But in contrast to Gauss elimination, we first apply a *row division* in order to make the pivot equal to 1, and only after that eliminate the nonzero entries in column 1 (which then requires no further divisions):

$$\begin{array}{lcl}
 & A = \begin{bmatrix} \mathbf{2} & 4 & 2 & 2 & -2 \\ 6 & 12 & 6 & 7 & 1 \\ 4 & 8 & 2 & 2 & 6 \end{bmatrix} & (r=0) \\
\text{divide (row 1) by } 2: & \downarrow & \\
 & \begin{bmatrix} \mathbf{1} & 2 & 1 & 1 & -1 \\ 6 & 12 & 6 & 7 & 1 \\ 4 & 8 & 2 & 2 & 6 \end{bmatrix} & \\
\text{subtract } 6\cdot\text{(row 1) from (row 2):} & \downarrow & \\
 & \begin{bmatrix} \mathbf{1} & 2 & 1 & 1 & -1 \\ 0 & 0 & 0 & 1 & 7 \\ 4 & 8 & 2 & 2 & 6 \end{bmatrix} & \\
\text{subtract } 4\cdot\text{(row 1) from (row 3):} & \downarrow & \\
 & \begin{bmatrix} \mathbf{1} & 2 & 1 & 1 & -1 \\ 0 & 0 & 0 & 1 & 7 \\ 0 & 0 & -2 & -2 & 10 \end{bmatrix} &
\end{array}$$

As we have made a downward step, we increase r and move on to column 2:

$$\begin{bmatrix} 1 & 2 & 1 & 1 & -1 \\ 0 & \mathbf{0} & 0 & 1 & 7 \\ 0 & 0 & -2 & -2 & 10 \end{bmatrix} \quad (r=1)$$

In Gauss elimination we would now be in the ugly case where no row exchange can bring a nonzero entry into the pivot position. However, this is a good case now. There is simply no downward step in column 2, and we directly move on to column 3. In this column, it is possible to make a row exchange to get a nonzero pivot:

$$\begin{bmatrix}1&2&1&1&-1\\0&0&\mathbf{0}&1&7\\0&0&-2&-2&10\end{bmatrix}\quad (r=1)$$

exchange (row 2) and (row 3): $\downarrow$

$$\begin{bmatrix}1&2&1&1&-1\\0&0&\mathbf{-2}&-2&10\\0&0&0&1&7\end{bmatrix}$$

divide (row 2) by -2: $\downarrow$

$$\begin{bmatrix}1&2&1&1&-1\\0&0&\mathbf{1}&1&-5\\0&0&0&1&7\end{bmatrix}$$

Now we eliminate in column 3, but unlike in Gauss elimination, we now also eliminate *above* the pivot, since REF requires that the pivot columns become standard unit vectors.

$$\begin{bmatrix}1&2&1&1&-1\\0&0&\mathbf{1}&1&-5\\0&0&0&1&7\end{bmatrix}$$

subtract $1\cdot$(row 2) from (row 1): $\downarrow$

$$\begin{bmatrix}1&2&0&0&4\\0&0&\mathbf{1}&1&-5\\0&0&0&1&7\end{bmatrix}$$

No elimination below the pivot is necessary here, so our downward step is done, we increase r and move to column 4. We already have a pivot of 1 in the desired position, so we need no row exchange and no row division. All that is left to do in this downward step is one row subtraction:

$$\begin{bmatrix}1&2&0&0&4\\0&0&1&1&-5\\0&0&0&\mathbf{1}&7\end{bmatrix}\quad (r=2)$$

subtract $1\cdot$(row 3) from (row 2): $\downarrow$

$$R_0=\begin{bmatrix}1&2&0&0&4\\0&0&1&0&-12\\0&0&0&\mathbf{1}&7\end{bmatrix}$$

After increasing r, we now have $r=3$ (number of rows) already; further downward steps are neither possible nor necessary, so we can stop even before having looked at the last column. Indeed, the matrix R_0 that we have now is in $\mathrm{REF}(1,3,4)$, and even in $\mathrm{RREF}(1,3,4)$ as there are no zero rows in the end. But in general, the result of Gauss-Jordan elimination can be a matrix R_0 with some zero rows in the end. As a simple example, think of the same starting matrix A as above, with a zero row appended in the end.

Here is the algorithm in general.

Theorem 3.21 (Gauss-Jordan elimination)**.** *Let A be an $m\times n$ matrix. There exists an invertible $m \times m$ matrix M such that $R_0 = MA$ is in* REF.

The matrix M turns out to be the product of all row operation matrices that we use during elimination in order to transform the initial matrix A into the final matrix R_0. We can also compute M itself on the way, by transforming a second initial $m \times m$ identity matrix I via the same row operations. This will result in M as a second final matrix. In our example above, we get

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

You can check that indeed, $R_0 = MA$ for A in (3.9) and R_0 the result of elimination as obtained on the previous page.

Proof. We proceed by induction on n where the base case is $n = 0$. If you have not looked into Exercise 2.24, you may wonder what an $m \times 0$ matrix looks like, a matrix with no columns. But whatever it looks like, it is in $\mathrm{REF}()$ with $r = 0$ according to Definition 3.19, so $M = I$ works. Indeed, the definition in this case requires something "for $i = 1, 2, \ldots, r$" in (i) and "for all i, j" in (ii). This means that there are in fact no requirements, since there are no applicable i in (i) and no applicable pairs (i, j) in (ii); see also the remark after Definition 2.3. If you don't like the base case $n = 0$, you can also convince yourself that any $m \times 1$ matrix can be transformed to REF, for example by repeating the induction step below to handle this case.

Now we consider $n > 0$ and assume that the statement already holds for all $m \times (n-1)$ matrices. Then we can write A as

$$A = \left[\begin{array}{c|c} & | \\ \underbrace{A'}_{m\times(n-1)} & \mathbf{v} \\ & | \end{array} \right]$$

and use the induction hypothesis to get $R_0' = M'A'$ where R_0' is in REF and M' is invertible. Under the hood, the induction hypothesis has already performed elimination in all but the last column of A: We have

$$M'A = \left[\begin{array}{c|c} & | \\ \underbrace{R_0'}_{m\times(n-1)} & \mathbf{w} \\ & | \end{array} \right],$$

where $\mathbf{w} = M'\mathbf{v}$. Let R_0' be in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$. Then $M'A$ looks like in Figure 3.5 (left).

There are two cases: if $w_i = 0$ for all $i > r$, we are done; see Figure 3.5 (middle). In this case, $M'A$ is also in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$, so we can set $M = M'$ to conclude that $R_0 = MA$ is in REF. This case also happens if $r = m$ in which case there are no applicable $i > r$.

In the other case, there is some $i > r$ such that $w_i \neq 0$, indicated by $\star$ in Figure 3.5 (right). In this case, we do some more row operations to transform $M'A$ into REF. We

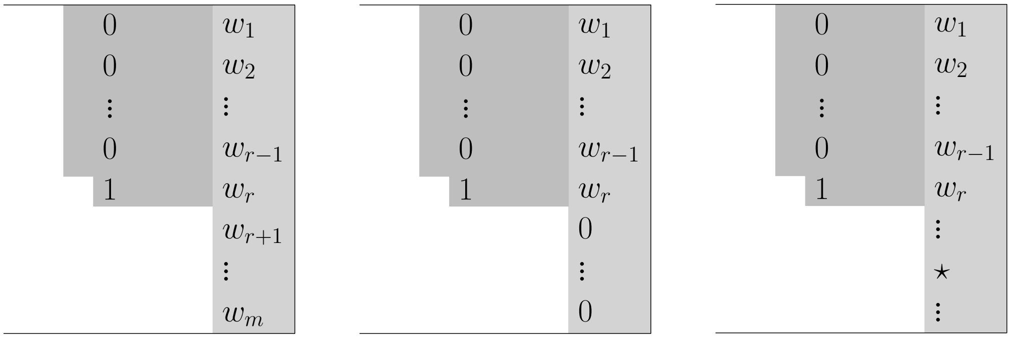


Figure 3.5: Matrix $M'A$ is in REF, except possibly the last column (left). There are two cases: $w_i = 0$ for all $i > r$ (middle), or there is some $i > r$ such that $w_i \neq 0$ (right).

first perform a row exchange (if necessary) to get $\star$ into row $r + 1$; see Figure 3.6 (left). This does not affect the previous columns since they have zero entries in rows $i > r$. Let M_1 be the corresponding (invertible) row operation matrix.

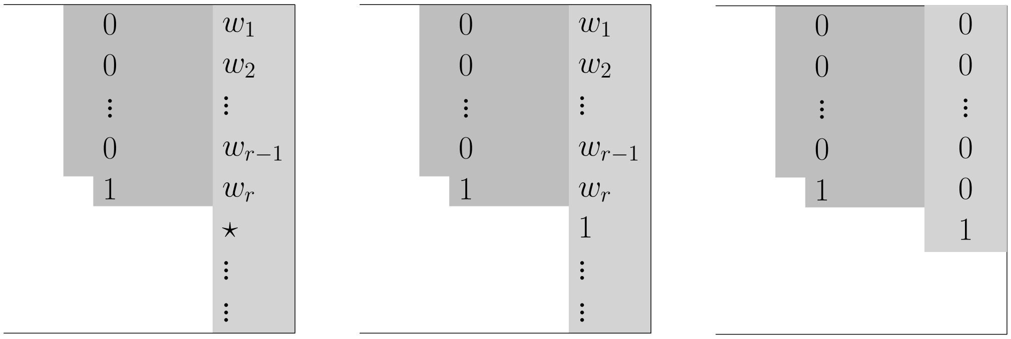


Figure 3.6: Performing row operations to transform $M'A$ into REF: a row exchange (left), a row division (middle), and eliminations in the last column (right)

Next, we perform a *row division* which divides a row by a nonzero scalar. Here, we divide row $r + 1$ by $\star$ so that $\star$ turns into 1; see Figure 3.6 (middle). Again, this does not change the previous columns, since $\star$ is the only nonzero entry in this row. Like all other row operations, a row division is undoable (multiply the row by $\star$!) and is therefore realized by an invertible row operation matrix M_2. Concretely, M_2 results from the $m \times m$ identity matrix by replacing the 1 in row $r + 1$ and column $r + 1$ with $1/\star$.

Finally, we perform eliminations: subtract suitable multiples of row $r+1$ from *all* other rows, also the ones above row $r + 1$, so that the last column is transformed into $\mathbf{e}_{r+1}$; see Figure 3.6 (right). As before, this leaves the previous columns untouched. Let M_3 be the product of all elimination matrices needed for the eliminations.

The matrix R_0 resulting after all the row operations is now in $\mathrm{REF}(j_1, j_2, \ldots, j_r, j_{r+1})$

with $j_{r+1} = n$ and satisfies

$$R_0 = \underbrace{M_3 M_2 M_1 M'}_{M} A = MA,$$

where M—as a product of invertible matrices—is invertible; see Lemma 3.9. □

From R_0 as obtained from A through Gauss-Jordan elimination, we can directly read off the independent columns of A.

Lemma 3.22. *Let A be an $m \times n$ matrix, M an invertible $m \times m$ matrix, and $R_0 = MA$ in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$. Then A has independent columns $j_1, j_2, \ldots, j_r$.*

Proof. We will argue that A and $R_0 = MA$ have their independent columns at the same positions. Then the statement follows since R_0 has independent columns $j_1, j_2, \ldots, j_r$, as shown in the proof of Observation 3.20.

The argument is a refinement of what we did in Corollary 3.4 to prove that multiplication with a row operation matrix does not affect linear independence of the columns.

Column j of A is independent if and only if there exists a vector $\mathbf{x} \in \mathbb{R}^n$ such that

$$A\mathbf{x} = \mathbf{0}, x_j = -1, x_k = 0 \text{ for } k > j.$$

Indeed, after adding the j-th column of A to both sides of $A\mathbf{x} = \mathbf{0}$, this expresses column j as a linear combination of the previous columns.

Similarly, column j of R_0 is independent if and only if there is $\mathbf{x} \in \mathbb{R}^n$ such that

$$R_0\mathbf{x} = \mathbf{0}, x_j = -1, x_k = 0 \text{ for } k > j.$$

Now, if such a vector exists for A, the same vector also works for $R_0 = MA$, and vice versa: since M is invertible, $A\mathbf{x} = \mathbf{0} \Leftrightarrow MA\mathbf{x} = \mathbf{0}$ (this is Lemma 3.3 applied with $\mathbf{b} = \mathbf{0}$). Hence, A and $R_0 = MA$ have their independent columns at the same positions. □

It is worthwhile to understand what Gauss-Jordan elimination does on an invertible $m \times m$ matrix A. In this case, all columns are independent, so by the previous Lemma, the resulting $m \times m$ matrix R_0 is in $\mathrm{REF}(1, 2, \ldots, m)$, and hence equal to the identity matrix I. So we have $R_0 = I = MA$, and this means that the matrix M in Theorem 3.21 is actually the inverse of A.

3.5.4 Runtime

We have shown that Gauss elimination on an invertible $m \times m$ matrix A needs time $O(m^3)$, see Theorem 3.6. For Gauss-Jordan elimination, we have two more parameters: n, the number of columns of A, and r, the rank of A. The following result gives the runtime depending on these three parameters. If $n = r = m$, we again get $O(m^3)$. Hence, using Gauss-Jordan elimination on an invertible $m \times m$ is only by a constant factor slower than Gauss elimination. It will be slower, since it does more work in this case: reduce A to the identity matrix I instead of an upper triangular matrix U.

Theorem 3.23. *Let A be an $m \times n$ matrix of rank r, and let $\mathbf{b} \in \mathbb{R}^m$.*

(i) Using Gauss-Jordan elimination, A can be transformed into $R_0 = MA$ in REF *as given by Theorem 3.21 in time $O(rmn + mn)$.*

(ii) By simultaneously transforming the $m \times m$ identity matrix using the same row operations, $M = MI$ can be computed in additional time $O(rm^2 + m^2)$.

(iii) Given M, the system $A\mathbf{x} = \mathbf{b}$ can be solved in time $O(m^2)$.

Proof. The analysis for (i) and (ii) is similar as for Gauss elimination (Section 3.2.4); except that we count the steps much less exactly—the big-O lets us get away with this. In each of the r downward steps, we update some matrix entries. As there are mn entries in A, the bound in (i) follows, where the extra mn account for handling the columns in which we do not make downward steps. The bound in (ii) is simply (i) applied with $n = m$.

For (iii), we compute the matrix-vector product $\mathbf{c} = M\mathbf{b}$ which can be done in $O(m^2)$ time; this is easy to see using the definitions of matrix-vector multiplication in Section 2.1.1. After this, we directly solve $R_0\mathbf{x} = \mathbf{c}$, see Section 3.5.2. Knowing the independent column indices $j_1, j_2, \ldots, j_r$ of R_0 from its shape, this can actually be done in time $O(m)$ which is dominated by $O(m^2)$. Now we also have a solution of $A\mathbf{x} = \mathbf{b}$, or we know that there is no solution. The two systems are equivalent, since $A\mathbf{x} = \mathbf{b}$ and $R_0\mathbf{x} = \mathbf{c}$ are obtainable from each other by premultiplication of both sides with M and M^{-1}, respectively. □

If $m \leq n$, then (ii) does not lead to much extra work. If $m > n$ (A is tall and skinny), however, this step may cost more time than (i). If r and n are both small, the extra work for computing the $m \times m$ matrix M is significant.

In this case, we can skip (ii) and solve $A\mathbf{x} = \mathbf{b}$ via a *replay* approach: remembering the row operations we did in the r downward steps in (i) and applying them again to $\mathbf{b}$, we only need $O(rm)$ time to transform $\mathbf{b}$ into $\mathbf{c} = M\mathbf{b}$ and $O(m)$ time to directly solve $R_0\mathbf{x} = \mathbf{c}$.

3.5.5 Computing the CR decomposition

In Theorem 2.23, we have introduced the CR decomposition of an $m \times n$ matrix A. This writes A as

$$A = \underbrace{C}_{m\times r}\underbrace{R}_{r\times n},$$

where r is the rank of A, the matrix C contains the independent columns of A, and column j of R tells us how column j of A can be expressed as a linear combination of the independent columns. We have also seen that there is a unique such matrix R. What we left open is how to systematically compute it. Here is the answer.

Theorem 3.24. *Let A be an $m \times n$ matrix and let $A = CR$ as in Theorem 2.23. Let $R_0 = MA$ in* $\mathrm{REF}(j_1, j_2, \ldots, j_r)$ *be the result of Gauss-Jordan elimination on A, see Theorem 3.21.*

Then R results from R_0 by removing the zero rows at the end (if there are any); in particular, R is in $\mathrm{RREF}(j_1, j_2, \ldots, j_r)$*, and C is the submatrix of A with columns $j_1, j_2, \ldots, j_r$.*

This may be a bit surprising. Upfront, it is not clear what CR decomposition has to do with Gauss-Jordan elimination. The theorem also implies that the result R_0 of Gauss-Jordan elimination is unique, despite the fact that there are some choices in the algorithm that could potentially influence the result. These choices concern the row exchanges. If in some step, there is more than one nonzero entry $\star$ below a zero pivot, then there is more than one way to bring a nonzero entry up. But apparently, it doesn't matter which entry we bring up. This is also easy to see directly if we review the proof of Theorem 3.21 with this aspect in mind.

Before we prove the theorem, let's check it on an example. In Section 2.2.3, we have considered

$$A = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 2 & 4 & 1 & 4 \\ 3 & 6 & 2 & 5 \end{bmatrix}$$

and manually computed

$$A = \underbrace{\begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 3 & 2 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & 0 & 1 & -2 \end{bmatrix}}_{R}.$$

To get R_0, we apply Gauss-Jordan elimination on A. This is a particularly simple case, since no row exchanges and row divisions are necessary, and also no eliminations above the pivot:

$$\begin{array}{rr}
 & A = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 2 & 4 & 1 & 4 \\ 3 & 6 & 2 & 5 \end{bmatrix} \\
\text{elimination in column 1:} & \downarrow \\
 & \begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 2 & -4 \end{bmatrix} \\
\text{elimination in column 3:} & \downarrow \\
 & R_0 = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\end{array}$$

After removing the zero row in the end, we indeed have R in this case.

Proof of Theorem 3.24. Let $R_0 = MA$ be in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$. Plugging in $A = CR$ gives

$$R_0 = MCR,$$

where C is the submatrix of A containing columns $j_1, j_2, \ldots, j_r$: these are the independent ones by Lemma 3.22. Hence, MC is the submatrix of $MA = R_0$ containing columns $j_1, j_2, \ldots, j_r$. By Definition 3.19 of REF, these columns are the unit vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r$.

Therefore,

$$R_0 = MCR = \underbrace{\left[\begin{array}{c} \underbrace{I}_{r\times r} \\ \hline \underbrace{0}_{(m-r)\times r} \end{array}\right]}_{MC} R = \underbrace{\left[\begin{array}{c} \underbrace{R}_{r\times n} \\ \hline \underbrace{0}_{(m-r)\times n} \end{array}\right]}_{R_0}$$

Since R_0 has exactly $m - r$ zero rows at the end (one for each row $i > r$), R is indeed R_0 without the zero rows in the end. $\square$

There was another question we asked in Section 2.2.3: what is the CR decomposition good for? We will shed some light on this in Section 4.3).

Chapter 4

The Four Fundamental Subspaces

4.1 Vector spaces

In this section, we finally reveal the truth about vectors: thinking of them as arrows in some space $\mathbb{R}^m$ provides an incomplete picture. We introduce the abstract concept of a vector space and see that the $\mathbb{R}^m$'s form just one species in a whole fauna of vector spaces (an important species, though). The abstraction also allows to view subspaces such as the column space of a matrix as vector spaces themselves.

So far, we have said that vectors are elements of some space $\mathbb{R}^m$, and for $m = 2, 3$, we have drawn them as arrows in the 2-dimensional plane or in 3-dimensional space. But this is by far not the full picture. So what is a vector, really?

To approach this, we resort to an analogy: Asking someone what a mammal is might bring up the answer "a cat, for example." A more elaborate answer may also include other examples such as dogs, whales, lions, and humans. But this kind of answer doesn't *define* a mammal. Let's ask Wikipedia what a mammal is. Here is the answer:[1]

> A *mammal* [...] is a vertebrate animal of the class **Mammalia**. Mammals are characterized by the presence of milk-producing mammary glands for feeding their young, [...]

This is not a definition of an individual mammal, but of the *class* of mammals: what is it that makes a mammal a mammal? This is what we need to know in order to really understand mammals. The *species* of cats is just one of many in the class of mammals.

Coming back to vectors: Saying that a vector is an element of some $\mathbb{R}^m$ is actually like saying that a mammal is a cat. Both statements are false, but in case of vectors, we can blame it on ignorance: so far, we simply haven't seen any "species" of vectors beyond the $\mathbb{R}^m$'s. But such species exist in other corners of the world of mathematics (we will discover one of them soon). Knowing this, the more fundamental question arises: what

[1] `https://en.wikipedia.org/wiki/Mammal`, accessed August 27, 2024

do they have in common? What is it that that makes a vector a vector? Here is the high-level answer, in the same spirit as the Wikipedia definition of a mammal:

> A vector is an element of a **vector space**. Vector spaces are characterized by the presence of two operations on their elements: vector addition and scalar multiplication.

Table 4.1 summarizes the situation.

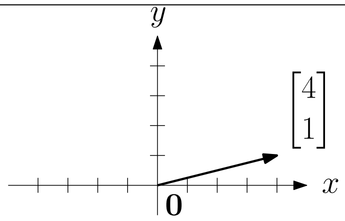
	vectors	mammals
individual		
breed	$\mathbb{R}^2$	Bombay cat
species	all $\mathbb{R}^m$'s	cats
class	vector spaces	Mammalia
characterization of the class	vector addition, scalar multiplication	milk-producing mammary glands

Table 4.1: Vectors vs. mammals

4.1.1 Definition and examples

After this high-level definition, here comes the actual definition of a *real vector space*. The word *real* doesn't stand for *true* or *proper*, but indicates that this is a vector space where the scalars are real numbers. Each $\mathbb{R}^m$ is a real vector space, but immediately after the definition, we will see a new "species." There are also vector spaces where the scalars are other kinds of numbers (*complex numbers* are an important case, and so are *bits*, the elements of the set $\{0, 1\}$), but we will not discuss them here. So we omit the word *real* and simply say vector space.

Don't be scared by the length of the definition; in particular, you will not be asked to learn the axioms by heart. You can look them up whenever needed.

Definition 4.1 (Vector space)**.** *A* vector space *is a triple $(V, +, \cdot)$ where V is a set (the vectors), and*

$$\begin{array}{rcll} + & : & V \times V \to V & \textit{is a function (vector addition),} \\ \cdot & : & \mathbb{R} \times V \to V & \textit{is a function (scalar multiplication),} \end{array}$$

satisfying the following axioms of a vector space *for all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ and all $\lambda, \mu \in \mathbb{R}$.*

1.	$\mathbf{v}+\mathbf{w}=\mathbf{w}+\mathbf{v}$	*commutativity*
2.	$\mathbf{u}+(\mathbf{v}+\mathbf{w})=(\mathbf{u}+\mathbf{v})+\mathbf{w}$	*associativity*
3.	*There is a vector* $\mathbf{0}$ *such that* $\mathbf{v}+\mathbf{0}=\mathbf{v}$ *for all* $\mathbf{v}$	*zero vector*
4.	*There is a vector* $-\mathbf{v}$ *such that* $\mathbf{v}+(-\mathbf{v})=\mathbf{0}$	*negative vector*
5.	$1\cdot\mathbf{v}=\mathbf{v}$	*identity element*
6.	$(\lambda\cdot\mu)\mathbf{v}=\lambda\cdot(\mu\cdot\mathbf{v})$	*compatibility of* $\cdot$ *and* $\cdot$ *in* $\mathbb{R}$
7.	$\lambda(\mathbf{v}+\mathbf{w})=\lambda\mathbf{v}+\lambda\mathbf{w}$	*distributivity over* $+$
8.	$(\lambda+\mu)\mathbf{v}=\lambda\mathbf{v}+\mu\mathbf{v}$	*distributivity over* $+$ *in* $\mathbb{R}$

Here, we use red color to indicate that there are two different "+", and also two different "$\cdot$". The red ones stand for the normal addition and multiplication of real numbers. The black ones are for vector addition and scalar multiplication. We will omit the coloring (and even the "$\cdot$") in the following, but there is (hopefully) only limited potential for confusion. After all, if you want to know which "+" or "$\cdot$" is meant in a given context, you simply need to check what is on the left side and the right side. We have done this kind of *overloading* before when we used the normal multiplication symbol "$\cdot$" also for the scalar product of two vectors, as in $\mathbf{v}\cdot\mathbf{w}$.

Now for the actual axioms: most of them seem obvious. Well, in $\mathbb{R}^m$, they are indeed (more or less) obvious, given how we have defined vector addition and scalar multiplication.

Observation 4.2. $(\mathbb{R}^m,+,\cdot)$, *with "+" as in Definition 1.1 and "$\cdot$" as in Definition 1.3, is a vector space.*

Previously, we have called this vector space $\mathbb{R}^m$ which—in hindsight—is an abuse of notation. But this is acceptable, since our "+" and "$\cdot$" are what mathematicians call the *canonical* (accepted standard) choices for vector addition and scalar multiplication in $\mathbb{R}^m$, so there is no strict need to mention them.

But in general, V could be any set, with $+$ and $\cdot$ defined in non-canonical ways, so we explicitly need to include these functions and also make sure that they behave as expected, where the expectations come from what happens in $\mathbb{R}^m$. This is what the axioms are for.

To illustrate the concept, we exhibit a new breed, the vector space of *real polynomials* in one variable. This belongs to the species of *polynomials* in several variables, but here we restrict to one variable. As for vector spaces, we omit the word *real*, since we do not consider any other polynomials in this course.

Definition 4.3 (Polynomial)**.** *A polynomial* $\mathbf{p}$ *is a sum of the form*

$$\mathbf{p}=\sum_{i=0}^{m}p_ix^i,$$

for some $m\in\mathbb{N}$. *Here* x *is a variable, and the numbers* $p_0,p_1,\ldots,p_m\in\mathbb{R}$ *are the* coefficients *of* $\mathbf{p}$. *The largest* i *such that* $p_i\neq 0$ *is the* degree *of* $\mathbf{p}$. *If all* p_i *are* 0, *we have the* zero polynomial $\mathbf{0}=0$ *whose degree we define to be* -1.

We note that m doesn't have to be the same for all polynomials, but for every polynomial, we have some m. For example, $\mathbf{p} = 2x^2 + x + 1$ is a polynomial of degree 2, and $\mathbf{q} = 5x - 2$ is a polynomial of degree 1.

To turn the set of polynomials into a vector space, we need to define addition of two polynomials, and multiplication of a polynomial with a scalar. There are canonical ways of doing this. In the example, we would define

$$\mathbf{p} + \mathbf{q} = 2x^2 + 6x - 1,$$

i.e. we add corresponding powers of x. And scalar multiplication simply scales all coefficients, as in

$$5\mathbf{p} = 10x^2 + 5x + 5.$$

Theorem 4.4. *Let $\mathbb{R}[x]$ be the set of polynomials in one variable x. Given polynomials $\mathbf{p} = \sum_{i=0}^{m} p_i x^i$ and $\mathbf{q} = \sum_{i=0}^{n} q_i x^i$, we define $\mathbf{p} + \mathbf{q}$ to be the polynomial*

$$\mathbf{p} + \mathbf{q} = \sum_{i=0}^{\max(m,n)} (p_i + q_i) x^i,$$

where we set $p_i = 0$ for $i > m$ and $q_i = 0$ for $i > n$. For a scalar $\lambda \in \mathbb{R}$, we further define $\lambda\mathbf{p}$ as the polynomial

$$\lambda\mathbf{p} = \sum_{i=0}^{m} (\lambda p_i) x^i.$$

Then $(\mathbb{R}[x], +, \cdot)$ is a vector space.

We omit the proof, since it is quite boring; it boils down to checking the obvious. Here is a second example: the vector space of $m \times n$ matrices. Again, we omit the easy but boring proof.

Theorem 4.5. *Let $\mathbb{R}^{m \times n}$ be the set of $m \times n$ matrices, with addition $A + B$ and scalar multiplication λA defined in the usual way, see Definition 2.2. Then $(\mathbb{R}^{m \times n}, +, \cdot)$ is a vector space.*

Proving the obvious. As boring as this may be, it is still surprising that we only need to check 8 "obvious" axioms to guarantee proper behavior of a vector space. Indeed, there are many other things that we expect from knowing how things work in $\mathbb{R}^m$. For example, we expect that there is only one zero vector in a vector space $(V, +, \cdot)$, but this doesn't appear among the axioms. So we need to prove that it follows from the axioms.

Fact 4.6. *Let $(V, +, \cdot)$ be a vector space. V contains exactly one zero vector (a vector satisfying axiom 3 of Definition 4.1: $\mathbf{v} + \mathbf{0} = \mathbf{v}$ for all $\mathbf{v}$).*

Proof. Take two zero vectors $\mathbf{0}$ and $\mathbf{0}'$. Then

$$\begin{aligned} \mathbf{0}' &= \mathbf{0}' + \mathbf{0} && \text{(by axiom 3, since } \mathbf{0} \text{ is a zero vector)} \\ &= \mathbf{0} + \mathbf{0}' && \text{(by axiom 1, commutativity)} \\ &= \mathbf{0} && \text{(by axiom 3, since } \mathbf{0}' \text{ is a zero vector).} \end{aligned}$$

So $\mathbf{0}$ and $\mathbf{0}'$ are equal. □

Doing this may feel a little bit like learning to walk again after a serious leg injury: hard work for something that we have previously taken for granted. Here is another such "relearning step."

Fact 4.7. *Let $(V, +, \cdot)$ be a vector space. For every $\mathbf{v} \in V$, there is exactly one negative vector $-\mathbf{v}$ (a vector satisfying axiom 4 of Definition 4.1: $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$).*

Proof. First of all, we need to realize that there is no way of simply computing $-\mathbf{v}$ as we do it in $\mathbb{R}^m$, by negating all entries. A vector $\mathbf{v}$ might not have any such "entries", and there is no "$-$" operator in $(V, +, \cdot)$ that we could apply. In the proof, we can only use the 8 axioms (and everything we have already derived from them). We could attempt to compute $-\mathbf{v}$ as $(-1)\mathbf{v}$ using scalar multiplication; this can be shown to produce a negative vector but does not rule out the existence of another negative vector. Here is how we go about that:

Take two negative vectors $\mathbf{u}$ and $\mathbf{u}'$ of $\mathbf{v}$. Then

$$\begin{aligned}
\mathbf{u}' &= \mathbf{u}' + \mathbf{0} && \text{(by axiom 3, zero vector)}\\
&= \mathbf{u}' + (\mathbf{v} + \mathbf{u}) && \text{(by axiom 4, since } \mathbf{u} \text{ is a negative of } \mathbf{v})\\
&= (\mathbf{u}' + \mathbf{v}) + \mathbf{u} && \text{(by axiom 2, associativity)}\\
&= (\mathbf{v} + \mathbf{u}') + \mathbf{u} && \text{(by axiom 1, commutativity)}\\
&= \mathbf{0} + \mathbf{u} && \text{(by axiom 4, since } \mathbf{u}' \text{ is a negative of } \mathbf{v}))\\
&= \mathbf{u} + \mathbf{0} && \text{(by axiom 1, commutativity)}\\
&= \mathbf{u} && \text{(by axiom 3, zero vector).}
\end{aligned}$$

So $\mathbf{u}$ and $\mathbf{u}'$ are equal. □

Having understood why a vector space is a triple $(V, +, \cdot)$ and not simply a set V of vectors, we will continue our (now more educated) abuse of notation and still write V for the vector space, with the understanding that vector addition and scalar multiplication are clear from the context. We also still write $\mathbf{0}$ for the zero vector when V is clear from the context.

4.1.2 Subspaces

Definition 4.8 (Subspace)**.** *Let V be a vector space. A nonempty subset $U \subseteq V$ is called a* subspace *of V if the following two* axioms of a subspace *are true for all $\mathbf{v}, \mathbf{w} \in U$ and all $\lambda \in \mathbb{R}$.*

(i) $\mathbf{v} + \mathbf{w} \in U$;

(ii) $\lambda\mathbf{v} \in U$.

These axioms guarantee that vector addition and scalar multiplication cannot take us out of the subspace. As a consequence, a subspace of V always contains at least the zero vector.

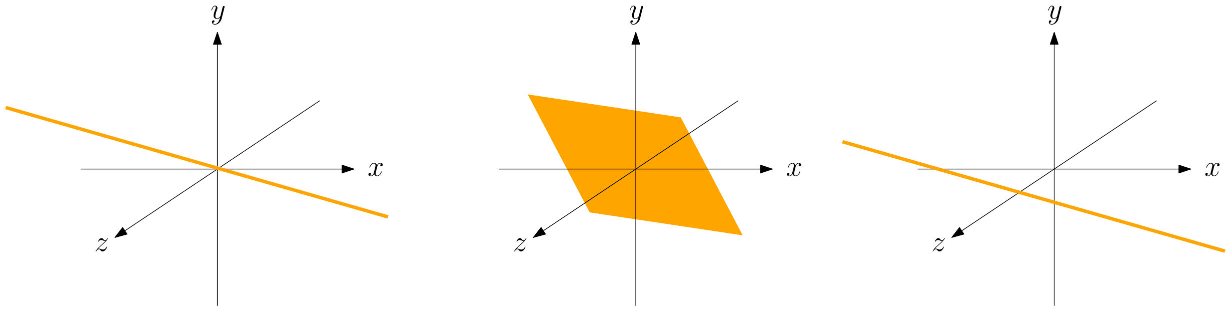


Figure 4.1: Subspaces of $\mathbb{R}^3$: a line through $\mathbf{0}$ (all scalar multiples of one vector); a plane through $\mathbf{0}$ (all linear combinations of two linearly independent vectors); the right subset is not a subspace, since it misses $\mathbf{0}$.

Lemma 4.9. *Let $U \subseteq V$ be a subspace of a vector space V. Then $\mathbf{0} \in V$.*

Proof. Take any $\mathbf{u} \in U$ (U is nonempty), By subspace axiom (ii), $0\mathbf{u} = \mathbf{0} \in U$. □

This proof seems quite obvious, but it is actually incomplete. While the equation $0\mathbf{u} = \mathbf{0}$ certainly holds in any $\mathbb{R}^m$, this does not automatically mean that it holds in all vector spaces. We need to prove it—another case of learning to walk again. We promise, it's the last one! In fact, the axioms of vector spaces have been carefully designed by mathematicians before us, with the goal of ensuring that everything that seems obvious is actually true. So we will relax and rely on this in the future.

Fact 4.10. *Let V be a vector space, $\mathbf{v} \in V$. Then $0\mathbf{v} = \mathbf{0}$.*

Proof.

$$\begin{array}{rll} & 0\mathbf{v} & \\ = & 0\mathbf{v} + \mathbf{0} & \text{(by axiom 3 of Definition 4.1, zero vector)} \\ = & 0\mathbf{v} + (0\mathbf{v} + (-0\mathbf{v})) & \text{(by axiom 4, negative vector)} \\ = & (0\mathbf{v} + 0\mathbf{v}) + (-0\mathbf{v}) & \text{(by axiom 2, associativity)} \\ = & (0+0)\mathbf{v} + (-0\mathbf{v}) & \text{(by axiom 8, distributivity over + in } \mathbb{R}) \\ = & 0\mathbf{v} + (-0\mathbf{v}) & \text{(by the rules of } \mathbb{R}) \\ = & \mathbf{0} & \text{(by axiom 4, negative vector)} \end{array}$$

□

Figure 4.1 gives two examples and one counterexample of subspaces of $\mathbb{R}^3$. Here is a subspace of $\mathbb{R}^m$ that we have already encountered, without thinking about it as a subspace: the column space of a matrix.

Lemma 4.11. *Let A be an $m \times n$ matrix. Then $\mathbf{C}(A) = \{A\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\}$ is a subspace of $\mathbb{R}^m$.*

Proof. Let $\mathbf{v}, \mathbf{w}$ be in $\mathbf{C}(A)$. Then there exist vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ such that $\mathbf{v} = A\mathbf{x}, \mathbf{w} = A\mathbf{y}$. Hence,

$$A(\underbrace{\mathbf{x}+\mathbf{y}}_{\in\mathbb{R}^n}) = A\mathbf{x} + A\mathbf{y} = \mathbf{v} + \mathbf{w} \quad \Rightarrow \quad \mathbf{v} + \mathbf{w} \in \mathbf{C}(A).$$

This was subspace axiom (i). For axiom (ii), let $\lambda \in \mathbb{R}$. Then

$$A(\underbrace{\lambda\mathbf{x}}_{\in\mathbb{R}^n}) = \lambda A\mathbf{x} = \lambda\mathbf{v} \quad \Rightarrow \quad \lambda\mathbf{v} \in \mathbf{C}(A).$$

In both chain of equalities, the first equality comes from the interpretation of matrix-vector multiplication as a linear transformation of the vector; see Observation 2.26. □

Knowing that a subspace always contains $\mathbf{0}$, we can actually say more.

Lemma 4.12. *Let V be a vector space and U a subspace. Then U is also a vector space (with the same "+" and "·" as V).*

Proof. Formally, to turn "+" and "·" into functions that work for U, we have to restrict their domains to $U \times U$ and $\mathbb{R} \times U$, respectively. The subspace axioms (i) and (ii) then also restrict their ranges to U.

Next, we need to check the 8 axioms. All but axiom 4 are true for all vectors in V, since V is a vector space; in particular, they hold for all vectors in U, so there is nothing to check. In Case of axiom 3, we are also using that $\mathbf{0} \in U$ (Lemma 4.9). What remains is axiom 4: we need to make sure that for all $\mathbf{u} \in U$, $-\mathbf{u}$ is actually in U; so far we only know that it is in V. But this holds, since $(-1)\mathbf{u} \in U$ by subspace axiom (ii), and "obviously" $(-1)\mathbf{u} = -\mathbf{u}$. If you are up for it, you can prove the obvious, otherwise, you can safely believe it. □

Subspaces of $\mathbb{R}[x]$. Let's look at some subspaces of $\mathbb{R}[x]$, the vector space of polynomials (Theorem 4.4). A polynomial *without constant term* is a polynomial of the form

$$\mathbf{p} = \sum_{i=0}^{m} p_i x^i \text{ where } p_0 = 0.$$

An example is $x^2 + 3x$. It is clear that these polynomials form a subspace of $\mathbb{R}[x]$, since the sum of two polynomials without constant term is again a polynomial without constant term, and so is each scalar multiple of a polynomial without constant term.

A *quadratic polynomial* is a polynomial $\mathbf{p}$ of the form

$$\mathbf{p} = p_0 + p_1 x + p_2 x^2.$$

We do not require $p_2 \neq 0$, so $3x$ is also a quadratic polynomial. Again, it is easy to see that the quadratic polynomials form a subspace of $\mathbb{R}[x]$. In fact, this subspace looks a lot like

$\mathbb{R}^3$: each quadratic polynomial $\mathbf{p}$ is determined by three real numbers p_0, p_1, p_2, so we can also describe it by a vector

$$\mathbf{v_p} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix} \in \mathbb{R}^3.$$

Moreover, "+" and "$\cdot$" on quadratic polynomials translate to "+" and "$\cdot$" on the corresponding vectors:

$$\mathbf{v_{p+q}} = \mathbf{v_p} + \mathbf{v_q}, \quad \mathbf{v_{\lambda p}} = \lambda \mathbf{v_p}.$$

Therefore, the subspace of quadratic polynomials is just $\mathbb{R}^3$ in disguise. The mathematical term is that the two spaces are *isomorphic*.

This allows us an interesting view of $\mathbb{R}[x]$ as the *union* of all $\mathbb{R}^m, m = 0, 1, \ldots$. In this union, we can also add vectors of different dimensions, for example

$$\underbrace{\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}}_{2x^2+x^2+1} + \underbrace{\begin{bmatrix} -2 \\ 5 \end{bmatrix}}_{5x-2} = \underbrace{\begin{bmatrix} -1 \\ 6 \\ 2 \end{bmatrix}}_{2x^2+6x-1}.$$

In this sense, polynomials form a "super breed" containing $\mathbb{R}^m$ for all m.

Subspaces of $\mathbb{R}^{m\times n}$. Let's turn to $\mathbb{R}^{m\times n}$, the vector space of matrices (Theorem 4.5). Here, we first observe that this is not really a new breed of vectors spaces, as $\mathbb{R}^{m\times n}$ is isomorphic to $\mathbb{R}^{mn}$: an $m \times n$ matrix is one way of grouping mn numbers, an mn-dimensional vector is another way. In both cases, vector addition and scalar-multiplication are defined entry-wise, so it doesn't really matter how we group the numbers.

The difference is really in how we think about vectors and matrices. A vector in $\mathbb{R}^{mn}$ is a "flat" 1-dimensional array, while a matrix in $\mathbb{R}^{m\times n}$ is a 2-dimensional array of rows and columns; see also Section 3.1.2 on computer vectors and matrices.

The 2-dimensional view leads to subspaces that would not make intuitive sense in a 1-dimensional array. For the examples, we consider $\mathbb{R}^{2\times 2}$.

Our first subspace is the set of symmetric matrices, the ones of the form

$$\begin{bmatrix} a & b \\ b & d \end{bmatrix}.$$

To see that this is a subspace, it suffices to observe that the sum of two symmetric matrices is symmetric, and that scaling a symmetric matrix keeps it symmetric. The second and slightly more creative example are the matrices of *trace* 0, the ones of the form

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \text{ where } a + d = 0.$$

Generally, the trace of a square matrix is the sum of the diagonal elements. Again, it is easy to see that the subspace axioms are satisfied. However, matrices of trace 1 do not form a subspace, and there are many other non-subspaces, for example the invertible matrices, the ones of rank 1, etc. Try to find violations of the subspace axioms for all of them!

4.2 Bases and dimension

The dimension of a vector space is an important measure of its complexity. So far, we only have an intuitive understanding of dimension, according to which $\mathbb{R}^m$ has dimension m. In this section, we first define bases of vector spaces and prove via the Steinitz exchange lemma that all bases have the same size. This allows us to define the dimension of a vector space as the size of any basis of it. Moroever, a basis provides a compact description of the vector space; computing a vector space means to compute a basis of it.

From working with the vector spaces $\mathbb{R}^m$, we have an intuitive understanding of dimension. According to this, $\mathbb{R}^m$ has dimension m. But if V is some other vector space, we may not have such an intuition, so we need to *define* the dimension of a vector space. We expect this definition to tell us that $\mathbb{R}^m$ indeed has dimension m.

But for example, what is the dimension of the vector space of polynomials introduced in Section 4.1.1)? As it "contains" $\mathbb{R}^m$ for all m, we expect the dimension to be infinite.

4.2.1 Bases

The crucial concept here is that of a *basis*. A basis of a vector space V consists of linearly independent vectors whose span is V (see Section 1.3.3 for the definition of span). Previously, we have defined linear independence and span for a *sequence* of vectors, and only in $\mathbb{R}^m$. This was important to handle for example the sequence of columns of a matrix that may contain duplicates. But here, *sets* of vectors turn out to be more practical. So we start by recalling the definitions of linear combination, linear independence and span, adapted for a set of vectors in a general vector space. At the same time, we extend them such that we can also handle infinite sets.

Definition 4.13 (Linear combination of a set of vectors)**.** *Let V be a vector space, $G \subseteq V$ a (possibly infinite) subset of vectors. A* linear combination *of G is a sum of the form*

$$\sum_{\mathbf{v}\in F} \lambda_{\mathbf{v}}\mathbf{v},$$

where $F \subseteq G$ is a finite subset of G and $\lambda_{\mathbf{v}} \in \mathbb{R}$ for all $\mathbf{v} \in F$.

Summing over all elements of a set as in $\sum_{\mathbf{v}\in F} \lambda_{\mathbf{v}}\mathbf{v}$ is in general not well-defined, since a set does not have an order of its elements. But if the result of the summation is the same for every possible order, this notation can be used. Here, this is the case, since vector addition is commutative (axiom 1 in Definition 4.1).

A linear combination can also be written in the usual way, namely as $\sum_{\mathbf{v}\in G} \lambda_{\mathbf{v}}\mathbf{v}$, and requiring that only finitely many $\lambda_{\mathbf{v}}$ are nonzero.

Here is an important fact. It is obvious in any $\mathbb{R}^m$, but for a general vector space, we have to prove it.

Lemma 4.14. *Let V be a vector space, $G \subseteq V$. Every linear combination of G is again in V.*

Proof. Let $\sum_{\mathbf{v}\in F} \lambda_{\mathbf{v}}\mathbf{v}$ be a linear combination, $|F| = n$. Enumerating the elements of F in arbitrary order $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, we can write the combination as $\sum_{j=1}^{n} \lambda_j \mathbf{v}_j$.

Since V is a vector space, we have $\mathbf{w}_j := \lambda_j \mathbf{v}_j \in V$ for all j, by definition of the scalar multiplication (function $\cdot : \mathbb{R} \times V \to V$). By definition of vector addition (function $+ : V \times V \to V$), we also have $\mathbf{w}_1 + \mathbf{w}_2 \in V$. Applying this again yields $(\mathbf{w}_1 + \mathbf{w}_2) + \mathbf{w}_3 \in V$, and so on, until we get the desired conclusion $\mathbf{w}_1 + \mathbf{w}_2 + \cdots + \mathbf{w}_n \in V$. (Under the hood, this is a proof by induction, and it uses the "obvious" fact that brackets can be omitted in writing down a sum of vectors). □

You may wonder why we do not allow infinite linear combinations. Infinite sums in themselves can be defined, you may for example know the formula

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x} \quad \text{for } x < 1.$$

The problem is that Lemma 4.14 may fail for infinite linear combinations, and we don't want this. Consider the vector space of polynomials $\mathbb{R}[x]$, and let G be the infinite subset of *unit monomials*: $1, x^2, x^3, \ldots$. With $\lambda_{\mathbf{p}} = 1$ for all $\mathbf{p} \in G$, we get the infinite "linear combination"

$$\sum_{\mathbf{p}\in G} \lambda_{\mathbf{p}}\mathbf{p} = \sum_{i=0}^{\infty} x^i.$$

This is *not* a polynomial. The vector space axioms only imply that *finite* linear combinations of vectors are again vectors, but infinite ones may be undefined or take us out of the vector space, as we have just seen.

With linear combinations as in Definition 4.13, we can now define span and linear independence in the usual way.

Definition 4.15 (Span and linear independence of sets of vectors)**.** *Let V be a vector space, $G \subseteq V$ a (possibly infinite) subset of vectors.*

The span *of G, written as* $\mathbf{Span}(G)$, *is the set of all linear combinations of G. The set G is called* linearly independent *if no vector $\mathbf{v} \in G$ is a linear combination of $G \setminus \{\mathbf{v}\}$.*

Now we can formally define a basis.

Definition 4.16 (Basis)**.** *Let V be a vector space. A subset $B \subseteq V$ of vectors is called a* basis *of V if B is linearly independent and* $\mathbf{Span}(B) = V$.

Examples. The set $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$ of standard unit vectors (Section 1.2.2) is a basis of $\mathbb{R}^m$. For example, if $m = 2$, then

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

These two vectors are linearly independent and span $\mathbb{R}^2$: for every vector

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \in \mathbb{R}^2,$$

we have $\mathbf{v} = v_1\mathbf{e}_1 + v_2\mathbf{e}_2$. For general m, the standard unit vectors are seen to be linearly independent by the *private nonzero* argument: every standard unit vector has a nonzero entry (a 1-entry, actually) at a coordinate where all other standard unit vectors have 0-entries. We call such an entry a *private nonzero*. A vector with a private nonzero cannot be a linear combination of the other ones, and if every vector has a private nonzero, the vectors are linearly independent.

Lemma 4.17. *Let A be an $m \times n$ matrix. The set of independent columns of A (Definition 2.9) is a basis of the column space $\mathbf{C}(A)$.*

Proof. $\mathbf{C}(A)$ is a subspace by Lemma 4.11. The independent columns are in the column space and linearly independent: by definition, no independent column is a linear combination of the previous columns, and this means that the independent columns are in fact linearly independent; see Corollary 1.20. Furthermore, the independent columns span the column space, as we have shown in Lemma 2.10. □

For the subspace of symmetric 2×2 matrices

$$\begin{bmatrix} a & b \\ b & d \end{bmatrix},$$

the following set of three symmetric matrices is a basis.

$$\left\{ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right\}.$$

They are linearly independent: every matrix has at least one *private nonzero*, a 1-entry where all other matrices have 0-entries. It remains to observe that every symmetric matrix is a linear combination of these three matrices:

$$\begin{bmatrix} a & b \\ b & d \end{bmatrix} = a \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + b \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + d \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

For the trace-0 matrices

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}, a + d = 0,$$

a basis is

$$\left\{\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}, \begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}\right\}.$$

Linear independence is again easy due to *private nonzero*s, and as $d = -a$ in a trace-0 matrix, we can obtain every trace-0 matrix as a linear combination:

$$\begin{bmatrix}a & b\\c & -a\end{bmatrix} = a\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} + b\begin{bmatrix}0 & 1\\0 & 0\end{bmatrix} + c\begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}.$$

For the vector space $\mathbb{R}[x]$ of polynomials, the infinite set of unit monomials

$$\{x^i : i = 0, 1, \ldots\}$$

is a basis. By Definition 4.3, every polynomial is a linear combination of unit monomials (observe that $x^0 = 1$). It remains to argue that the unit monomials are linearly independent. Indeed, every unit monomial x^i has its *private nonzero*, the i-th power of x and can therefore not be obtained as a linear combination of other monomials.

The subspace of polynomials without a constant term has

$$\{x^i : i = 1, 2, \ldots\}$$

as a basis, and for the subspace of quadratic polynomials, a basis is

$$\{1, x, x^2\}.$$

Finally, what is the basis of $\{\mathbf{0}\}$, the smallest possible subspace of a given vector space? It is the empty set . Indeed, this is linearly independent by Definition 4.15: in the empty set, no vector is a linear combination of the others. And $\mathbf{Span}(\emptyset) = \{\mathbf{0}\}$, since an empty sum yields $\mathbf{0}$; see the discussion in Section 1.1.5.

There are typically many bases. The above examples should not trick us into believing that there is always only one basis of a vector space. For example, $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$ is the *canonical basis* of $\mathbb{R}^m$, but there are many other choices.

Observation 4.18. *Every set $B = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ of m linearly independent vectors is a basis of $\mathbb{R}^m$.*

Proof. B is linearly independent, so we only need to show that $\mathbf{Span}(B) = \mathbb{R}^m$, meaning that every vector $\mathbf{v} \in \mathbb{R}^m$ is a linear combination of B. For this, let A be the $m \times m$ matrix with (linearly independent) columns $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$. By Theorem 3.11, $A\mathbf{x} = \mathbf{v}$ has a unique solution $\mathbf{x}$, and

$$\mathbf{v} = \underbrace{\sum_{j=1}^{m} x_j \mathbf{v}_j}_{A\mathbf{x}}$$

is the desired linear combination of B. □

As a second example, let us consider the column space $\mathbf{C}(A)$ of a matrix A. In Section 2.2.3, we have computed the independent columns of

$$A = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 2 & 4 & 1 & 4 \\ 3 & 6 & 2 & 5 \end{bmatrix}.$$

These are the first and the third column, so these two columns form a basis of $\mathbf{C}(A)$. But we could also have defined the "backwards independent" columns, by going through the columns of A from *right to left*, and selecting a column if it is not a linear combination of the ones *succeeding* it. This also results in a basis of $\mathbf{C}(A)$, by the same arguments as for the "forward independent" columns. If we do this in our example, we end up with the fourth and the third column as a basis.

More generally, we could go through the columns in *any* order and pick up the ones that are not linear combinations of the ones previously considered. This potentially gives us many different bases of $\mathbf{C}(A)$.

However, what we find in both $\mathbb{R}^m$ and $\mathbf{C}(A)$ is that the alternative bases still have the *same number* of vectors: m in case of $\mathbb{R}^m$ and 2 in case of the column space example. In the next section, we prove that this is not a coincidence but true for every vector space.

4.2.2 The Steinitz exchange lemma

Let V be a vector space, and suppose that $F \subseteq V$ is a finite set of linearly independent vectors, and $G \subseteq V$ a finite set of vectors with $\mathbf{Span}(G) = V$. The Steinitz exchange lemma makes two statements. The first one is that $|F| \leq |G|$. This makes sense: In $\mathbb{R}^2$, for example, we can have *at most* 2 linearly independent vectors, and it takes *at least* 2 vectors to span $\mathbb{R}^2$.

The second stament sounds a bit technical: we can enlarge F by elements from G such that the enlarged set has at most the size of G and also spans V. The name of the lemma comes from the fact that we can think of this as "exchanging elements between G and F".

Lemma 4.19 (Steinitz exchange lemma). *Let V be a vector space, $F \subseteq V$ a finite set of linearly independent vectors, and $G \subseteq V$ a finite set of vectors with $\mathbf{Span}(G) = V$. Then the following two statements hold.*

(i) $|F| \leq |G|$.

(ii) There exists a subset $E \subseteq G$ of size $|G| - |F|$ such that $\mathbf{Span}(F \cup E) = V$.

Note that the set E in (ii) is allowed to contain elements of F. In this case, $|F \cup E| < |G|$. There is a stronger version of the lemma in which we require $E \subseteq G \setminus F$, and this always leads to $|F \cup E| = |G|$. But we will not need this.

Proof. We adapt the proof from Wikipedia.[2] The proof is by induction on $f = |F|$. The base case is $f = 0$ in which case F is the empty set. Then (i) is clear and for (ii), we chose $E = G$.

If $f > 0$, pick an arbitrary vector $\mathbf{u} \in F$, set $F' = F \setminus \{\mathbf{u}\}$, and let $g = |G|$. Note that F' is also linearly independent. By the induction hypothesis, we have

(i) $g \geq f - 1$.

(ii) There exists a subset $E' \subseteq G$ of size $g - (f - 1)$ with $\mathbf{Span}(F' \cup E') = V$.

Since $\mathbf{u} \in V = \mathbf{Span}(F' \cup E')$, there are scalars $\lambda_{\mathbf{v}}, \mathbf{v} \in F' \cup E'$ such that

$$\mathbf{u} = \sum_{\mathbf{v} \in F' \cup E'} \lambda_{\mathbf{v}} \mathbf{v}. \tag{4.1}$$

There must be some $\mathbf{w} \in E'$ with $\lambda_{\mathbf{w}} \neq 0$. Indeed, since F is linearly independent, $\mathbf{u}$ is not a linear combination of $F' = F \setminus \{\mathbf{u}\}$ (Definition 4.15). This shows that $|E'| = g - (f - 1) \geq 1$, so we get $g \geq f$ which already proves (i) for size f.

To show (ii) for size f, we remove $\mathbf{w}$ from E' to obtain E. This set has the required size $g - f$, and it remains to prove that $\mathbf{Span}(F \cup E) = V$.

Towards this, we solve (4.1) for $\mathbf{w}$:

$$\mathbf{w} = \frac{1}{\mu_{\mathbf{w}}} \left(\mathbf{u} - \sum_{\mathbf{v} \in F' \cup E} \lambda_{\mathbf{v}} \mathbf{v} \right).$$

Thus, $\mathbf{w}$ is a linear combination of $\{u\} \cup F' \cup E = F \cup E$, and by Lemma 1.23, this means that

$$\mathbf{Span}(F \cup E) = \mathbf{Span}(\underbrace{F \cup E \cup \{\mathbf{w}\}}_{F \cup E'}). \tag{4.2}$$

Formally, we would need a version of Lemma 1.23 for general vector spaces, and for finite sets instead of finite sequences. But we skip this (the proof would be the same as before).

Recall from (4.1) that $\mathbf{u}$ is a linear combination of $F' \cup E'$, so Lemma 1.23 also gives

$$\mathbf{Span}(F' \cup E') = \mathbf{Span}(\underbrace{F' \cup E' \cup \{\mathbf{u}\}}_{F \cup E'}). \tag{4.3}$$

Putting together (4.2) and (4.3), and using the inductive hypothesis, we get

$$\mathbf{Span}(F \cup E) = \mathbf{Span}(F' \cup E') = V.$$

□

The Steinitz exchange lemma has an important corollary. Because of its importance, we also call it a theorem. It confirms what we have observed in examples before: even if a vector space has different bases, all of them have the same number of vectors.

[2] `https://en.wikipedia.org/wiki/Steinitz_exchange_lemma`, accessed August 5, 2024

Theorem 4.20. *Let V be a vector space and $B, B' \subseteq V$ two finite bases of V. Then $|B| = |B'|$.*

Proof. By Definition 4.16 of a basis, B and B' are linearly independent, and $\mathbf{Span}(B) = \mathbf{Span}(B') = V$. Then, statement (i) of the Steinitz exchange lemma with $F = B, G = B'$ yields $|B| \leq |B'|$; with $F = B', G = B$, we get $|B'| \leq |B|$. □

There are vector spaces that do not have finite bases, such as the vector space $\mathbb{R}[x]$ of polynomials defined in Section 4.1.1. While the theorem as presented here does not apply to such vector spaces, it can be generalized to the infinite case where $|B| = |B'|$ then means "the same kind of infinity."

A more fundamental question is this: does every vector space even have a basis? The answer is yes, even in the infinite case. Proving this involves some machinery that is standard but beyond the scope of these notes. The Wikipedia article about bases of vector spaces is a good entry point for further reading.[3] Here we will present a proof for the finite case. This case is defined as follows.

Definition 4.21 (Finitely generated vector space)**.** *A vector space V is called* finitely generated *if there exists a finite subset $G \subseteq V$ with $\mathbf{Span}(G) = V$.*

For example, $\mathbb{R}^m$ is finitely generated (by $G = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$) but $\mathbb{R}[x]$, the vector space of polynomials, is not.

Theorem 4.22. *Let V be a finitely generated vector space, and let $G \subseteq V$ be a finite subset with $\mathbf{Span}(G) = V$. Then V has a basis $B \subseteq G$.*

Proof. This is what we call an "algorithmic proof'." It constructs B by an algorithm. Here is how it goes.

"line 1"

If G is linearly independent, $B = G$ is a basis by Definition 4.16. Otherwise, there is a vector $\mathbf{v} \in G$ that is a linear combination of the other vectors, so we have $\mathbf{Span}(G \setminus \{\mathbf{v}\}) = \mathbf{Span}(G) = V$ via Lemma 1.23. Then we replace G with $G \setminus \{\mathbf{v}\}$ (which still spans V) and go back to line 1. As G gets smaller in every step, this must eventually stop and produce a basis. □

A formal correctness proof would go via a precise formulation of the algorithm, either as a loop (correctness proof with *loop invariants*), or a *recursive algorithm* (correctness proof by induction). The latter approach is very close to directly proving the theorem by induction on $g = |G|$ (this is a good exercise). Our proof above, while hopefully being clear and easy to understand, is of a somewhat informal "and-so-on" nature, but knowing how we could make it formal if necessary, this is acceptable.

4.2.3 Dimension

Now we can define the dimension of a vector space, at least if it is finitely generated.

[3] https://en.wikipedia.org/wiki/Basis_(linear_algebra), accessed August 5, 2024

Definition 4.23 (Dimension). *Let V be a finitely generated vector space. Then $\dim(V)$, the dimension of V, is the size of any basis B of V.*

This definition uses that every finitely generated vector space has a basis to begin with (Theorem 4.22), and that all bases have the same size (Theorem 4.20).

From the examples of bases in Section 4.2.1, you can therefore immediately deduce the dimensions of the corresponding vector spaces. As expected, $\dim(\mathbb{R}^m) = m$.

Figure 4.2 shows three subspaces of $\mathbb{R}^3$, of dimensions 0 (point), 1 (line), and 2 (plane).

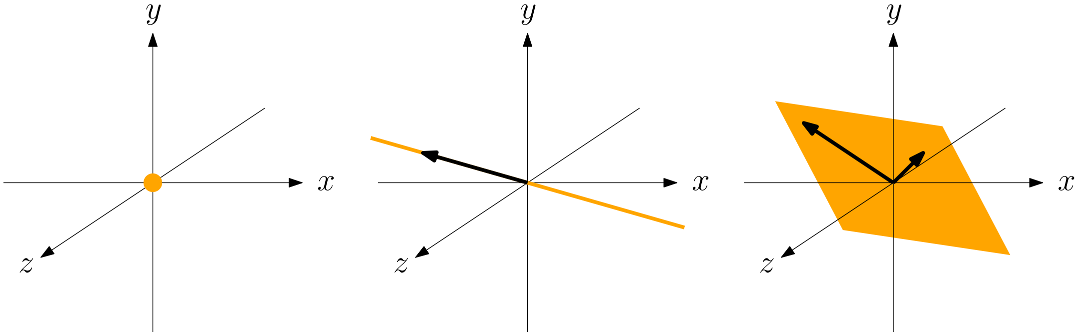


Figure 4.2: Three subspaces of $\mathbb{R}^3$: The unique subspace of dimension 0—the point at the origin with an empty basis (left); a subspace of dimension 1—a line through the origin with a basis of size 1 (middle); a subspace of dimension 2—a plane through the origin with a basis of size 2 (right)

We also have the following result that simplifies basis checks: if $\dim(V)$ many vectors are linearly independent, we already know that their span is V, and the other way around. In both cases, the set can therefore be identified as a basis, from only one of the basis axioms in Definition 4.16.

Lemma 4.24. *Let V be a vector space with $\dim(V) = d$.*

(i) Let $F \subseteq V$ be a set of d linearly independent vectors. Then F is a basis of V.

(ii) Let $G \subseteq V$ be a set of d vectors with $\mathbf{Span}(G) = V$. Then G is a basis of V.

Proof. For (i), let G be a basis of V. Since $\mathbf{Span}(G) = V$, the Steinitz exchange Lemma 4.19 applies with F and G, but since $|F| = |G| = d$, the set E in part (ii) can only be the empty set. Hence, $\mathbf{Span}(F) = \mathbf{Span}(F \cup E) = V$, and F is a basis according to Definition 4.16.

For (ii), we use Theorem 4.22 which guarantees a basis $B \subseteq G$ of size d. But since $|B| = |G| = d$, we must have $B = G$, so G itself is a basis. □

4.3 Computing the fundamental subspaces

An $m \times n$ matrix A defines four fundamental subspaces. Column space and row space are two of them, and here we introduce the other two: nullspace and left nullspace. We show how they can be computed by which we mean to find bases for them. Once we have such bases, we also know the dimensions of the fundamental subspaces, and how they relate to each other: If $r = \mathbf{rank}(A)$, both row and column space have dimension r, while the nullspace has dimension $n - r$, and the left nullspace has dimension $m - r$. We finally apply these results to understand the space of all solutions of a system of linear equations $A\mathbf{x} = \mathbf{b}$.

A vector space V typically contains infinitely many vectors. But if V is finitely generated, we can compute a basis of it. Such a basis is a finite representation of V in the sense that it allows us to describe all elements of V: these are simply all linear combinations of the basis. Moreover, for every $\mathbf{v} \in V$, there is a unique such linear combination by (the vector space version of) Lemma 1.21. By *computing a vector space*, we mean to compute a basis of it.

Figure 4.2 visualizes this representation by a basis: knowing the basis vectors, the space in question is uniquely determined.

For a given $m \times n$ matrix A, we now want to compute the four *fundamental subspaces* $\mathbf{C}(A)$ (column space), $\mathbf{R}(A)$ (row space), $\mathbf{N}(A)$ (nullspace) and $\mathbf{LN}(A)$ (left nullspace). If we have bases for them, we also know the dimensions of these spaces, by Definition 4.23. Column and row space have been introduced before (Definitions 2.8 and 2.13), but here the focus is on looking at them at subspaces of $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively.

The key to understanding these subspaces is Gauss-Jordan elimination. In Section 3.5.3, we have shown how this algorithm can transform every matrix A into a matrix R_0 in row echelon form (REF). As our running example, we use the one considered in Sections 2.2.3 and 3.5.5 before:

$$A = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 2 & 4 & 1 & 4 \\ 3 & 6 & 2 & 5 \end{bmatrix} \quad \rightarrow \quad R_0 = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.4}$$

From this, we have already been able to read off the CR decomposition of A that we have previously computed manually (Section 2.2.3). Now we will see how to read off (bases of) the fundamental subspaces.

4.3.1 Column space

We recall Definition 2.8. The column space of an $m \times n$ matrix A is the set of all linear combinations of the columns of A,

$$\mathbf{C}(A) = \{A\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\} \subseteq \mathbb{R}^m.$$

The column space is sometimes also denoted by $\mathbf{Im}(A)$ and called the *image* of A. This is because $\mathbf{C}(A)$ is the image of the linear transformation $T(\mathbf{x}) = A\mathbf{x}$; see Definition 2.31.

We know from Lemma 4.11 that $\mathbf{C}(A)$ is a subspace. For computing it, we have already done all the work. We summarize the situation in the following theorem.

Theorem 4.25. *Let A be an $m \times n$ matrix, and let R_0 in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$ be the result of Gauss-Jordan elimination on A, according to Theorem 3.21. Then A has independent columns $j_1, j_2, \ldots, j_r$, and these form a basis of the column space $\mathbf{C}(A)$. Hence*

$$\dim(\mathbf{C}(A)) = r = \mathbf{rank}(A).$$

Proof. The independent columns of A form a basis of $\mathbf{C}(A)$ by Lemma 4.17, and using Gauss-Jordan elimination, we can compute their positions; see Lemma 3.22, telling us that A has independent columns $j_1, j_2, \ldots, j_r$. □

In the running example (4.4), R_0 is in $\mathrm{REF}(1, 3)$, so the basis B of $\mathbf{C}(A)$ consists of the two independent columns 1 and 3,

$$B = \left\{ \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \right\}.$$

4.3.2 Row space

The row space of an $m \times n$ matrix A is the set of all linear combinations of the rows of A; to avoid concept repetition, we have officially defined the row space as the column space of the transpose, see Definition 2.13:

$$\mathbf{R}(A) = \mathbf{C}(A^\top) \subseteq \mathbb{R}^n.$$

From Lemma 4.11 applied to $A^\top$, we immediately obtain

Corollary 4.26. *Let A be an $m \times n$ matrix. Then $\mathbf{R}(A)$ is a subspace of $\mathbb{R}^n$.*

We could compute $\mathbf{R}(A) = \mathbf{C}(A^\top)$ via Gauss-Jordan elimination on $A^\top$, according to Theorem 4.25. But the nice thing is that our previous Gauss-Jordan elimination on A already provides us with all we need. The following lemma is the key.

Lemma 4.27. *Let A be an $m \times n$ matrix and M an invertible $m \times m$ matrix. Then $\mathbf{R}(A) = \mathbf{R}(MA)$.*

Proof. We need to prove that $\mathbf{C}(A^\top) = \mathbf{C}((MA)^\top)$. Since $(MA)^\top = A^\top M^\top$ (Lemma 2.19) and $N := M^\top$ is also invertible (Lemma 3.10), we set $B := A^T$ and prove $\mathbf{C}(B) = \mathbf{C}(BN)$

as follows:

$$\begin{array}{c}
\mathbf{v} \in \mathbf{C}(B) \\
\Updownarrow \\
\mathbf{v} = B\mathbf{x} \text{ for some } \mathbf{x} \in \mathbb{R}^m \\
\Uparrow \qquad\qquad \Downarrow \qquad\qquad \leftarrow \mathbf{y} := N^{-1}\mathbf{x} \quad \Leftrightarrow \quad \mathbf{x} := N\mathbf{y} \\
\mathbf{v} = BN\mathbf{y} \text{ for some } \mathbf{y} \in \mathbb{R}^m \\
\Updownarrow \\
\mathbf{v} \in \mathbf{C}(BN).
\end{array}$$

□

Using this, a basis of $\mathbf{R}(A)$ can be found as follows.

Theorem 4.28. *Let A be an $m \times n$ matrix, and let R_0 in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$ be the result of Gauss-Jordan elimination on A, according to Theorem 3.21. Then the first r rows of R_0 form a basis of the row space $\mathbf{R}(A)$, and*

$$\dim(\mathbf{R}(A)) = r = \mathbf{rank}(A).$$

Proof. Gauss-Jordan elimination according to Theorem 3.21 provides us with a matrix $R_0 = MA$ in REF and the same row space as A, by Lemma 4.27. From R_0, a basis of the row space can be read off immediately: R_0 starts with r nonzero rows and ends with $m - r$ zero rows. Hence, the r nonzero rows already span the row space, and they are also linearly independent, as each of them has a *private nonzero* (where the "downward step" occurs). Finally, $r = \mathbf{rank}(A)$ holds by Lemma 3.22. □

In our running example (4.4), the basis B is

$$B = \left\{ \begin{bmatrix} 1 & 2 & 0 & 3 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & -2 \end{bmatrix} \right\}.$$

This theorem has a very interesting consequence that also deserves to be called a theorem, although it is now simply a corollary of Theorems 4.25 and 4.28. These two theorems together imply that $\dim(\mathbf{C}(A)) = \dim(\mathbf{R}(A))$. Since $\mathbf{R}(A) = \mathbf{C}(A^\top)$, we also get $\dim(\mathbf{C}(A)) = \dim(\mathbf{C}(A^\top))$, meaning that both A and $A^\top$ have the same number of independent columns, hence the same rank.

This is the following result that is sometimes summarized as *row rank equals columns rank*.

Theorem 4.29. *Let A be an $m \times n$ matrix. Then*

$$\mathbf{rank}(A) = \mathbf{rank}(A^\top).$$

We have previously proved this in Section 2.1.4 for matrices of rank 1, but now we know that it holds for all matrices.

We also obtain a particularly beautiful interpretation of the CR decomposition.

Corollary 4.30. *Let $A = CR$ be the CR decomposition of A (Section 2.2.3). By Theorem 4.25, the columns of C form a basis of the column space $\mathbf{C}(A)$. By Theorem 4.28 together with Theorem 3.24, the rows of R form a basis of the row space $\mathbf{R}(A)$. Both spaces have the same dimension $r = \mathbf{rank}(A) = \mathbf{rank}(A^\top)$.*

4.3.3 Nullspace

We now get to the third fundamental subspace defined by a matrix A.

Definition 4.31 (Nullspace). *Let A be an $m \times n$ matrix. The* nullspace *of A is the set of all solutions of $A\mathbf{x} = \mathbf{0}$,*

$$\mathbf{N}(A) = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{0}\} \subseteq \mathbb{R}^n.$$

The nullspace is sometimes also denoted by $\mathbf{Ker}(A)$ and called the *kernel* of A. The reason is that $\mathbf{N}(A)$ is the kernel of the linear transformation $T(\mathbf{x}) = A\mathbf{x}$; see Definition 2.31.

Lemma 4.32. *Let A be an $m \times n$ matrix. Then $\mathbf{N}(A)$ is a subspace of $\mathbb{R}^n$.*

Proof. Let $\mathbf{v}, \mathbf{w} \in \mathbf{N}(A)$, $\lambda \in \mathbb{R}$. As in the proof of Lemma 4.11 we use that A realizes a linear transformation and directly get the subspace axioms in Definition 4.8.

$$A(\mathbf{v} + \mathbf{w}) = \underbrace{A\mathbf{v}}_{\mathbf{0}} + \underbrace{A\mathbf{w}}_{\mathbf{0}} = \mathbf{0} \quad \Rightarrow \quad \mathbf{v} + \mathbf{w} \in \mathbf{N}(A)$$

and

$$A(\lambda\mathbf{v}) = \lambda \underbrace{A\mathbf{v}}_{\mathbf{0}} = \mathbf{0} \quad \Rightarrow \quad \lambda\mathbf{v} \in \mathbf{N}(A).$$

□

Gauss-Jordan elimination also reveals the nullspace of A, with the help of the following lemma. This is actually Lemma 3.3 "reloaded", with $\mathbf{b} = \mathbf{0}$; back then, we did not know inverse matrices yet and therefore had to argue with undoing a row operation.

Lemma 4.33. *Let A be an $m \times n$ matrix and M an invertible $m \times m$ matrix. Then A and MA have the same nullspace $\mathbf{N}(A) = \mathbf{N}(MA)$.*

Proof. For $\mathbf{x} \in \mathbb{R}^n$, we have

$$\begin{array}{ccccccc}
\mathbf{x} \in \mathbf{N}(A) & \Leftrightarrow & A\mathbf{x} = \mathbf{0} & \Rightarrow & MA\mathbf{x} = M\mathbf{0} & & \\
 & & \Uparrow & & \Downarrow & & \\
 & & M^{-1}MA = M^{-1}\mathbf{0} & \Leftarrow & MA\mathbf{x} = \mathbf{0} & \Leftrightarrow & \mathbf{x} \in \mathbf{N}(MA).
\end{array}$$

□

To show how to extract a basis of $\mathbf{N}(A) = \mathbf{N}(R_0)$ from the matrix $R_0 = MA$ that we obtain from Gauss-Jordan elimination, we look at the running example (4.4) first. Here,

$$R_0 = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

We observe that $\mathbf{N}(R_0) = \mathbf{N}(R)$ where

$$R = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & 0 & 1 & -2 \end{bmatrix}$$

is obtained from R_0 by removing the zero row at the end. Indeed, the system $R\mathbf{x} = \mathbf{0}$ is equivalent to $R_0\mathbf{x} = \mathbf{0}$, since the extra equations in $R_0\mathbf{x} = \mathbf{0}$ are of the form "$0 = 0$" and therefore always hold. Recall from the previous Section 4.3.2 that R is the matrix in the CR decomposition $A = CR$, and that the rows of R form a basis of the row space $\mathbf{R}(A)$.

In the running example, the matrix R is in $\mathrm{RREF}(1,3)$, hence we can write $R\mathbf{x} = \mathbf{0}$ as

$$\underbrace{\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}}_{I}\underbrace{\begin{bmatrix}x_1\\x_3\end{bmatrix}}_{\mathbf{x}(I)} + \underbrace{\begin{bmatrix}2 & 3\\0 & -2\end{bmatrix}}_{Q}\underbrace{\begin{bmatrix}x_2\\x_4\end{bmatrix}}_{\mathbf{x}(Q)} = \mathbf{0}.$$

The submatrix of R with the independent columns $1, 3$ is the identity matrix, and we let Q be the submatrix of R with the dependent columns $2, 4$. Furthermore, $\mathbf{x}(I)$ and $\mathbf{x}(Q)$ denote the subvectors of $\mathbf{x}$ "belonging" to I and Q.

We therefore can further write $R\mathbf{x} = \mathbf{0}$ as

$$\mathbf{x}(I) = -Q\mathbf{x}(Q).$$

Solving this system is very easy: the *free variables* $\mathbf{x}(Q)$ can be substituted with arbitrary real numbers. The values of the *basic variables* $\mathbf{x}(I)$ are then determined via $\mathbf{x}(I) = -Q\mathbf{x}(Q)$. We have already encountered this "direct solution" approach in Section 3.5.2 for solving $A\mathbf{x} = \mathbf{b}$ when A is in REF.

Here, the goal is different: we want to compute a basis of $\mathbf{N}(R) = \mathbf{N}(A)$. It turns out that there is a basis of *special solutions*, the ones where $\mathbf{x}(Q)$ is set to one of the standard unit vectors; see Table 4.2.

		special solutions	
free variables	$\underbrace{\begin{bmatrix}x_2\\x_4\end{bmatrix}}_{\mathbf{x}(Q)}$	$\underbrace{\begin{bmatrix}1\\0\end{bmatrix}}_{\mathbf{v}_1(Q)}$	$\underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\mathbf{v}_2(Q)}$
basic variables	$\underbrace{\begin{bmatrix}-2 & -3\\0 & 2\end{bmatrix}}_{-Q}\underbrace{\begin{bmatrix}x_2\\x_4\end{bmatrix}}_{\mathbf{x}(Q)} = \underbrace{\begin{bmatrix}x_1\\x_3\end{bmatrix}}_{\mathbf{x}(I)}$	$\underbrace{\begin{bmatrix}-2\\0\end{bmatrix}}_{\mathbf{v}_1(I)}$	$\underbrace{\begin{bmatrix}-3\\2\end{bmatrix}}_{\mathbf{v}_2(I)}$
nullspace equation	$\mathbf{0} = \underbrace{\begin{bmatrix}1 & 2 & 0 & 3\\0 & 0 & 1 & -2\end{bmatrix}}_{R}\underbrace{\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}}_{\mathbf{x}}$	$\underbrace{\begin{bmatrix}-2\\1\\0\\0\end{bmatrix}}_{\mathbf{v}_1}$	$\underbrace{\begin{bmatrix}-3\\0\\2\\1\end{bmatrix}}_{\mathbf{v}_2}$

Table 4.2: The special solutions $\mathbf{v}_1$ and $\mathbf{v}_2$ form a basis of the nullspace $\mathbf{N}(R) = \mathbf{N}(A)$.

Let's double check on the running example (4.4) that $\mathbf{v}_1, \mathbf{v}_2$ as computed in Table 4.2

are indeed in $\mathbf{N}(A)$. For this, we verify that

$$\underbrace{\begin{bmatrix}1&2&0&3\\2&4&1&4\\3&6&2&5\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}-2\\1\\0\\0\end{bmatrix}}_{\mathbf{v}_1}=\mathbf{0}, \quad \text{and} \quad \underbrace{\begin{bmatrix}1&2&0&3\\2&4&1&4\\3&6&2&5\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}-3\\0\\2\\1\end{bmatrix}}_{\mathbf{v}_2}=\mathbf{0}.$$

But why is $\{\mathbf{v}_1, \mathbf{v}_2\}$ a basis of $\mathbf{N}(R) = \mathbf{N}(A)$ according to Definition 4.16? This follows from the general argument that we make next. The notation with the double indices might appear a bit frightening, but from the example before, it should be quite clear what happens.

Lemma 4.34. *Let R be an $r \times n$ matrix in $\mathrm{RREF}(j_1, j_2, \ldots, .j_r)$, meaning that R has independent columns $j_1 < j_2 < \cdots < j_r$ that are equal to the standard unit vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r$ (Definition 3.19). We let $j_{r+1} < j_{r+2} < \cdots < j_n$ denote the indices of the dependent columns.*

The $r \times r$ submatrix of R formed by the independent columns is the identity matrix I. We let Q denote the $r \times (n-r)$ submatrix of R formed by the dependent columns.

For a vector $\mathbf{x} \in \mathbb{R}^n$, we let $\mathbf{x}(I) \in \mathbb{R}^r$ and $\mathbf{x}(Q) \in \mathbb{R}^{n-r}$ denote the subvectors

$$\mathbf{x}(I) = \begin{bmatrix}x_{j_1}\\x_{j_2}\\\vdots\\x_{j_r}\end{bmatrix} \in \mathbb{R}^r \text{ and } \mathbf{x}(Q) = \begin{bmatrix}x_{j_{r+1}}\\x_{j_{r+2}}\\\vdots\\x_{j_n}\end{bmatrix} \in \mathbb{R}^{n-r}$$

of basic and free entries.

Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n-r} \in \mathbb{R}^n$ be the $n-r$ vectors defined via

$$\mathbf{v}_i(Q) = \mathbf{e}_i \text{ and } \mathbf{v}_i(I) = -Q\mathbf{v}_i(Q), \quad i = 1, 2, \ldots, n-r. \tag{4.5}$$

Then $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r\}$ is a basis of $\mathbf{N}(R)$.

Proof. By construction, all vectors satisfy the nullspace equation $R\mathbf{x} = \mathbf{0}$ in its equivalent version $\mathbf{x}(I) = -Q\mathbf{x}(Q)$, so they are actually in the nullspace.

We further need to check that the vectors are linearly independent which follows from the usual *private nonzero* entry argument. More concretely, already the subvectors $\mathbf{v}_i(Q)$ are linearly independent, as they are different standard unit vectors. The full vectors are then linearly independent as well.

Finally, we must show that every vector in $\mathbf{N}(R)$ is a linear combination of the $\mathbf{v}_i$'s. For this, take any vector $\mathbf{x} \in \mathbf{N}(R)$. We claim that

$$\mathbf{x} = \sum_{i=1}^{n-r} x_{j_{r+i}} \mathbf{v}_i \tag{4.6}$$

is the desired linear combination. For the subvector of free entries, this reads as

$$\mathbf{x}(Q) = \sum_{i=1}^{n-r} x_{j_{r+i}} \underbrace{\mathbf{v}_i(Q)}_{=\mathbf{e}_i \text{ by (4.5)}} \tag{4.7}$$

and is therefore obvious by definition of the subvector and the $\mathbf{v}_i$'s. For the subvector of basic variables, we compute

$$\mathbf{x}(I) \overset{R\mathbf{x}=\mathbf{0}}{=} -Q\mathbf{x}(Q) \overset{(4.7)}{=} -Q\left(\sum_{i=1}^{n-r} x_{j_{r+i}}\mathbf{v}_i(Q)\right) = \sum_{i=1}^{n-r} x_{j_{r+i}}\left(\underbrace{-Q\mathbf{v}_i(Q)}_{=\mathbf{v}_i(I) \text{ by (4.5)}}\right) = \sum_{i=1}^{n-r} x_{j_{r+i}}\mathbf{v}_i(I).$$

Together with (4.7), this shows the claim (4.6). In the third equality, we were able to "pull Q in" due to "matrix-vector multiplication = linear transformation"; this needs the generalization of Observation 2.26 to more vectors, as provided by Lemma 2.28. □

In summary, we obtain the following result.

Theorem 4.35. *Let A be an $m \times n$ matrix, and let R_0 in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$ be the result of Gauss-Jordan elimination on A, according to Theorem 3.21. Let R in $\mathrm{RREF}(j_1, j_2, \ldots, j_r)$ be the submatrix of R_0 consisting of the first r rows. The vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots .\mathbf{v}_{n-r}$ as constructed in Lemma 4.34 form a basis of $\mathbf{N}(A) = \mathbf{N}(R_0) = \mathbf{N}(R)$, and therefore,*

$$\dim(\mathbf{N}(A)) = n - r = n - \mathbf{rank}(A).$$

Proof. By Definition 3.19 of REF and RREF, R is in $\mathrm{RREF}(j_1, j_2, \ldots, j_r)$, so Lemma 4.34 applies and yields a basis of $\mathbf{N}(R)$ with $n - r$ vectors. As R_0 differs from R only by extra zero rows, we have $\mathbf{N}(R) = \mathbf{N}(R_0)$. Finally, $\mathbf{N}(R_0) = \mathbf{N}(A)$ follows from Lemma 4.33, using that $R_0 = MA$ with M invertible, according to Theorem 3.21. As before, $r = \mathbf{rank}(A)$ holds by Lemma 3.22. □

4.3.4 Left nullspace

Just like the row space of a matrix A is defined as the column space of the transpose $A^\top$, the left nullspace of A is defined as the nullspace of $A^\top$.

Definition 4.36 (Left nullspace)**.** *Let A be an $m \times n$ matrix. The* left nullspace *of A is the set of all (row vector) solutions of $\mathbf{y}A = \mathbf{0}^\top$, or equivalently, the (column vector) solutions of $A^\top\mathbf{y} = \mathbf{0}$, hence the nullspace of $A^\top$:*

$$\mathbf{LN}(A) := \mathbf{N}(A^\top) \subseteq \mathbb{R}^m.$$

This is called *left* nullspace, since it contains the solutions of a system of equations $\mathbf{y}A = \mathbf{0}$ where the (row) vector of variables is multiplied with A from the left.

Lemma 4.32 applied to $A^\top$ immediately yields

Lemma 4.37. *Let A be an $m \times n$ matrix. Then $\mathbf{LN}(A)$ is a subspace of $\mathbb{R}^m$.*

Computing the left nullspace via Gauss-Jordan elimination on $A^\top$ as described in the previous Section 4.3.3 is a natural way to proceed, but as for the row space, there is a way to derive a basis directly from Gauss-Jordan elimination on A. This involves the matrix M, the "memory" of the row operations being performed.

Theorem 4.38. *Let A be an $m \times n$ matrix, and let $R_0 = MA$ in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$ be the result of Gauss-Jordan elimination on A according to Theorem 3.21. Then the last $m - r$ rows $\mathbf{w}_{r+1}, \mathbf{w}_{r+2}, \ldots, \mathbf{w}_m$ of the $m \times m$ matrix M form a basis of $\mathbf{LN}(A)$, and therefore,*

$$\dim(\mathbf{LN}(A)) = m - r = m - \mathbf{rank}(A).$$

Proof. Since R_0 ends with $m - r$ zero rows, the last $m - r$ rows of the matrix equation $R_0 = MA$ read as

$$0 = \begin{bmatrix} — & \mathbf{w}_{r+1} & — \\ — & \mathbf{w}_{r+2} & — \\ & \vdots & \\ — & \mathbf{w}_m & — \end{bmatrix} A.$$

This shows that $\mathbf{w}_{r+1}, \mathbf{w}_{r+2}, \ldots, \mathbf{w}_m \in \mathbf{LN}(A)$. Moreover, these vectors are linearly independent: Since M is invertible, its columns are linearly independent (Theorem 3.11), so $\mathbf{rank}(M) = m$; therefore also $\mathbf{rank}(M^\top) = m$ (Theorem 4.29), so the rows of M are also linearly independent.

It remains to show that $\mathbf{w}_{r+1}, \mathbf{w}_{r+2}, \ldots, \mathbf{w}_m$ is a basis. By applying Theorem 4.35 to $A^\top$, we already know $\dim(\mathbf{LN}(A)) = \dim(\mathbf{N}(A^\top)) = m - \mathbf{rank}(A^\top) = m - \mathbf{rank}(A) = m - r$. Here we have used Theorem 4.29 together with the fact that A has rank r due to R_0 in $\mathrm{REF}(j_1, j_2, \ldots, j_r)$.

Hence, we have found $m - r$ linearly independent vectors in the subspace $\mathbf{LN}(A)$ of dimension $m - r$. So these vectors must be a basis by Lemma 4.24 (i). □

Let's check this in our running example (4.4) which provides the initial matrix A and the final matrix R_0 in $\mathrm{REF}(1, 3)$. We still need the matrix M. You can verify that it is

$$M = \begin{bmatrix} 0 & 2 & -1 \\ 0 & -3 & 2 \\ 1 & -2 & 1 \end{bmatrix}.$$

From R_0, we know that $\mathbf{rank}(A) = 2$, so $\dim(\mathbf{LN}(A)) = 3 - 2 = 1$, and the last row of M forms a basis of $\mathbf{LN}(A)$:

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 & 3 \\ 2 & 4 & 1 & 4 \\ 3 & 6 & 2 & 5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}.$$

4.3.5 The solution space of $A\mathbf{x} = \mathbf{b}$

Having introduced and computed the four fundamental subspaces of a matrix A, we finally come back to solving systems of linear equations. Using Gauss elimination and back substitution (Sections 3.2.2 and 3.2.1), we have seen how to compute the unique solution $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$ when A is an invertible square matrix.

In the general case, we have applied Gauss-Jordan elimination and direct solution (Sections 3.5.3 and 3.5.2) to either find some solution, or to conclude that there is no solution. Here, we want to understand and compute the space of all solutions.

Definition 4.39 (Solution space). *Let A be an $m \times n$ matrix and $\mathbf{b} \in \mathbb{R}^m$. The set*

$$\mathbf{Sol}(A, \mathbf{b}) := \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}\} \subseteq \mathbb{R}^n$$

is the solution space *of* $A\mathbf{x} = \mathbf{b}$.

If $\mathbf{b} \neq \mathbf{0}$, $\mathbf{Sol}(A, \mathbf{b})$ is not a subspace of $\mathbb{R}^n$, simply because it doesn't contain the zero vector (see xLemma 4.9). Let's look at the situation where A is a 2×1 matrix, so we have a system in one equation and two variables. As a concrete example, consider the system

$$2x + 3y = 6.$$

The solutions of this, all pairs (x, y) satisfying $2x + 3y = 6$, form a line in $\mathbb{R}^2$, not containing the origin; see Figure 4.3. Replacing the right-hand side 6 by 0 results in another line which contains the origin and *is* a subspace (the nullspace of the matrix $\begin{bmatrix}2 & 3\end{bmatrix}$; see Section 4.3.3). In this example, we see that $\mathbf{Sol}(\begin{bmatrix}2 & 3\end{bmatrix}, [6])$ and $\mathbf{Sol}(\begin{bmatrix}2 & 3\end{bmatrix}, [0]) = \mathbf{N}(\begin{bmatrix}2 & 3\end{bmatrix})$ are parallel lines.

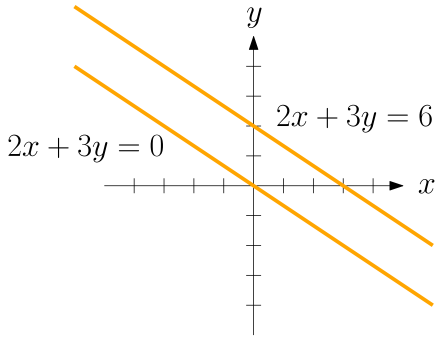


Figure 4.3: $\mathbf{Sol}(A, b)$ for $A = [2 \;\; 3], \mathbf{b} = [6]$ (upper line) and $\mathbf{b} = [0]$ (lower line).

Here is the general picture: For $\mathbf{b} \neq \mathbf{0}$, and if there is a solution at all, $\mathbf{Sol}(A, b)$ is obtained by "shifting" the nullspace of A "away from the origin."

Theorem 4.40. *Let A be an $m \times n$ matrix, $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{s}$ be some solution of $A\mathbf{x} = \mathbf{b}$. Then*

$$\mathbf{Sol}(A, \mathbf{b}) = \{\mathbf{s} + \mathbf{x} : \mathbf{x} \in \mathbf{N}(A)\}.$$

Hence, we can also compute $\mathbf{Sol}(A, b)$, despite the fact that it is not a subspace. To describe all solutions, we just need *some* solution (see Section 3.5.2) and a basis of $\mathbf{N}(A)$ (see Section 4.3.3).

Proof. We first show that every solution $\mathbf{y} \in \mathbf{Sol}(A, \mathbf{b})$ is of the form $\mathbf{s} + \mathbf{x}$ with $\mathbf{x} \in \mathbf{N}(A)$. Indeed, we can write $\mathbf{y}$ as

$$\mathbf{y} = \mathbf{s} + \underbrace{(\mathbf{y} - \mathbf{s})}_{\mathbf{x}},$$

and due to $A\mathbf{x} = A\mathbf{y} - A\mathbf{s} = \mathbf{b} - \mathbf{b} = \mathbf{0}$, we have $\mathbf{x} \in \mathbf{N}(A)$. For the other direction, we show that every vector $\mathbf{y}$ of the form $\mathbf{y} = \mathbf{s} + \mathbf{x}$ with $\mathbf{x} \in \mathbf{N}(A)$ is in $\mathbf{Sol}(A, \mathbf{b})$. For this, we compute

$$A\mathbf{y} = A\mathbf{s} + A\mathbf{x} = \mathbf{b} + \mathbf{0} = \mathbf{b}.$$

□

The number of solutions. A system $A\mathbf{x} = \mathbf{b}$ of linear equations has two characteristic numbers: m, the number of equations, and n, the number of variables. Then A is an $m \times n$ matrix. But if we want to understand the solution space, there is a third important characteristic number r, the rank of A. This is revealed only by Gauss-Jordan elimination on A: r is the number of "downward steps" in the row echelon form of the resulting matrix R_0; see Theorem 4.25.

If there is a solution at all, the solution space is obtained by shifting the nullspace $\mathbf{N}(A)$ away from the origin (Theorem 4.40). We can then declare $\mathbf{Sol}(A, \mathbf{b})$ to be of the same dimension as the nullspace, and this dimension is $n - r$ by Theorem 4.35. $\mathbf{Sol}(A, \mathbf{b})$ can be a point, a line, a plane,. . . , see Figure 4.4.

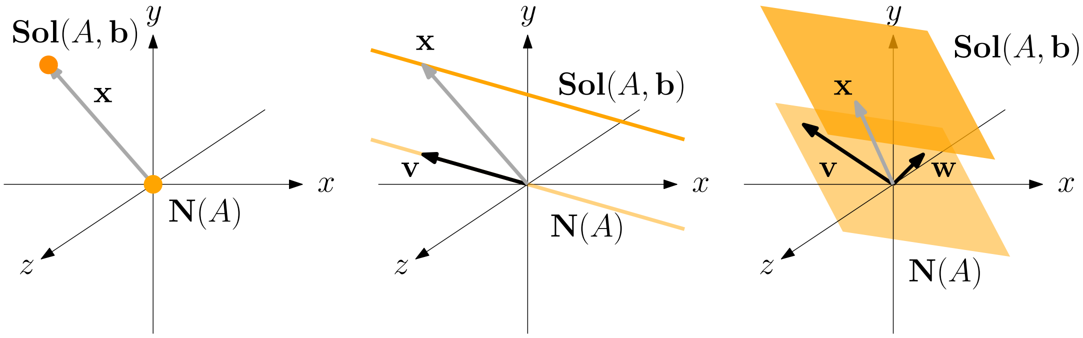


Figure 4.4: A solution space of dimension 0—a point (left); a solution space of dimension 1—a line (middle); a solution space of dimension 2—a plane (right)

Next, we want to understand in more detail in which cases there is a solution. As we will see, this typically depends on whether $r = m$ or not.

The case $r = m$ corresponds to the situation where the matrix R_0 resulting from Gauss-Jordan elimination (Theorem 3.21) is in reduced row echelon form RREF, with no zero rows at the end. We can arrive at this case if A is a square matrix, or a short and wide matrix ($n > m$), see Figure 2.1. In the latter case, we call the system $A\mathbf{x} = \mathbf{b}$ *underdetermined*.

If A is tall and skinny ($n < m$), the system is *overdetermined* and must have $r < m$. The reason is that r, the number of independent columns, cannot exceed n, the total number of columns.

Lemma 4.41. *Let A be an $m \times n$ matrix of rank $r = m$. Then $A\mathbf{x} = \mathbf{b}$ has a solution for every $\mathbf{b} \in \mathbb{R}^m$.*

Proof. From Theorem 4.25, we know that $\dim(\mathbf{C}(A)) = r = m$, so $\mathbf{C}(A)$ is a subspace of $\mathbb{R}^m$ of the same dimension m. Then we must have $\mathbf{C}(A) = \mathbb{R}^m$. This seems obvious, but actually needs an argument. The argument is that every basis B of $\mathbf{C}(A)$ is a set of m independent vectors in $\mathbb{R}^m$ and therefore also basis of $\mathbb{R}^m$ by Lemma 4.24 (i). Hence, $\mathbf{Span}(B) = \mathbf{C}(A) = \mathbb{R}^m$ by Definition 4.16 of a basis.

It follows that every $\mathbf{b} \in \mathbb{R}^m$ is in $\mathbf{C}(A)$, and this is the same as saying that $A\mathbf{x} = \mathbf{b}$ has a solution. □

In contrast, if $r < m$, the column space $\mathbf{C}(A)$ has dimension smaller than m. Then, "almost all" vectors $\mathbf{b} \in \mathbb{R}^m$ are not in $\mathbf{C}(A)$ and the system $A\mathbf{x} = \mathbf{b}$ is unsolvable. For an illustration, consider Figure 2.3 (right) where the column space is a line in $\mathbb{R}^2$. If we pick a "typical" $\mathbf{b}$ from $\mathbb{R}^2$ (whatever that precisely means), we expect $\mathbf{b}$ to land outside of that line. This can properly be formalized, and under this formalization, a subspace of $\mathbb{R}^m$ that has dimension smaller than m is a *set of measure* 0.

Hence, if we define a typical $\mathbf{b}$ to be one that is not in a given set of measure 0, then typical systems $A\mathbf{x} = \mathbf{b}$ are unsolvable if $r < m$. For example, overdetermined systems ($n < m$) are typically unsolvable. Underdetermined systems ($n > m$) are solvable (and then have infinitely many solutions) if we have a "typical" A, one of full rank $r = m$. To argue that short and wide matrices with $r < m$ are untypical and form a set of measure 0, we need *determinants* that will be introduced in the second part of the course. Similarly, square systems ($n = m$) are (uniquely) solvable for typical A.

Affine subspaces. Generally, a shifted copy of a subspace in a vector space is called an *affine subspace*. In the context of affine subspaces, a "normal" subspace is also sometimes called *linear subspace*. There is a full theory of *affine space*s which are essentially vector spaces without a distinguished origin.[4] The elements of an affine space are typically called *points*, and they are simply locations in space whose positions are "absolute" and not defined relative to a special origin $\mathbf{0}$.

4.3.6 Summary

Here we summarize the computations of the four fundamental subspaces of an $m \times n$ matrix A, and of the solution space $\mathbf{Sol}(A, \mathbf{b})$. Gauss-Jordan elimination (Theorem 3.21)

[4] `https://en.wikipedia.org/wiki/Affine_space`, accessed September 7, 2024

yields

$$MA = R_0,$$

where R_0 is in row echelon form $\mathrm{REF}(j_1, j_2, \ldots, j_r)$ for some $1 \le j_1 < j_2 < \cdots < j_r \le n$, and M is an invertible $m \times m$ matrix, recording all the row operations that happened during elimination. M and R_0 can be computed by performing Gauss-Jordan elimination *on the first n columns* of the extended $m \times (n + m)$ matrix $[A|I]$. This transforms A into $MA = R_0$ and I into $MI = M$. From M and R_0, bases of the four fundamental subspaces can be read off easily, see Figure 4.5 for our running example.

$$\mathbf{LN}(A) \qquad \mathbf{C}(A) \qquad \mathbf{N}(A) \qquad \mathbf{R}(A)$$

$$\underbrace{\begin{bmatrix}1\\0\end{bmatrix}}_{\mathbf{v}_1(Q)} \quad \underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\mathbf{v}_2(Q)}$$

$$\begin{matrix}\begin{bmatrix}1\\2\\3\end{bmatrix} & \begin{bmatrix}0\\1\\2\end{bmatrix} \\ \uparrow & \uparrow\end{matrix} \qquad \begin{matrix}\underbrace{\begin{bmatrix}-2\\0\end{bmatrix}}_{\mathbf{v}_1(I)} & \underbrace{\begin{bmatrix}-3\\2\end{bmatrix}}_{\mathbf{v}_2(I)} \\ \uparrow & \uparrow\end{matrix}$$

$$\begin{bmatrix}1 & -2 & 1\end{bmatrix} \leftarrow \underbrace{\begin{bmatrix}0 & 2 & -1\\0 & -3 & 2\\1 & -2 & 1\end{bmatrix}}_{M} \underbrace{\begin{bmatrix}1 & 2 & 0 & 3\\2 & 4 & 1 & 4\\3 & 6 & 2 & 5\end{bmatrix}}_{A} = \underbrace{\begin{bmatrix}1 & 2 & 0 & 3\\0 & 0 & 1 & -2\\0 & 0 & 0 & 0\end{bmatrix}}_{R_0 \text{ in REF(1,3)}} \underbrace{\begin{matrix}\to \\ \to \\ \end{matrix}\begin{bmatrix}1 & 2 & 0 & 3\\0 & 0 & 1 & -2\end{bmatrix}}_{R \text{ in RREF(1,3)}}$$

Figure 4.5: Computation of (bases of) the four fundamental subspaces of A, based on Gauss-Jordan elimination ($MA = R_0$)

Column space. Columns $j_1, j_2, \ldots, j_r$ of A (the independent columns) form a basis of the column space $\mathbf{C}(A)$; see Theorem 4.25.

Row space. The first r rows of R_0 (the nonzero rows) define a matrix R in reduced row echelon form $\mathrm{RREF}(j_1, j_2, \ldots, j_r)$. This matrix coincides with the matrix R in the CR decomposition of A; see Theorem 3.24. Moreover, the rows of R are a basis of the row space $\mathbf{R}(A)$; see Theorem 4.28.

Nullspace. For $i = 1, 2, \ldots, n - r$, the i-th basis vector $\mathbf{v}_i$ is obtained by setting the subvector $\mathbf{v}_i(Q)$ of the $n - r$ free variables (the ones corresponding to the dependent columns) to the i-th standard unit vector $\mathbf{e}_i$. The subvector $\mathbf{v}_i(I)$ of basic variables is then computed via $\mathbf{v}_i(I) = -Q\mathbf{v}_i(Q) = -Q\mathbf{e}_i$ where Q is the submatrix of R containing the $n - r$ dependent column; see Theorem 4.35. Note that $-Q\mathbf{e}_i$ is simply the negative of the i-th column of Q.

Left nullspace. The last $m - r$ rows of M form a basis of the left nullspace $\mathbf{LN}(A)$; see Theorem 4.38.

Table 4.3 summarizes the resulting dimensions (number of basis vectors) of the four subspaces.

subspace		
name	symbol	dimension
column space	$\mathbf{C}(A)$	r
row space	$\mathbf{R}(A)$	r
nullspace	$\mathbf{N}(A)$	$n - r$
left nullspace	$\mathbf{LN}(A)$	$m - r$

Table 4.3: Dimensions of the four fundamental subspaces of an $m \times n$ matrix A of $\mathbf{rank}(A) = r$.

Solution space of $A\mathbf{x} = \mathbf{b}$. The solution space $\mathbf{Sol}(A, \mathbf{b})$ is a "shifted copy" of the nullspace (Theorem 4.40) and therefore also of dimension $n - r$, *if* there is a solution. For $r = m$, this is always the case (Lemma 4.41) but for $r < m$, it is typically not the case (see the discussion after Lemma 4.41).

Bibliography

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117, 1998. `https://www.sciencedirect.com/science/article/pii/S016975529800110X`.

[CLRS22] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., and Stein, Clifford. *Introduction to Algorithms, Fourth Edition*. The MIT Press, 2022. `https://mitpress.mit.edu/9780262046305/introduction-to-algorithms/`.

[CSB06] V. Claus, A. Schwill, and R. Böving. *Duden Informatik A - Z: Fachlexikon für Studium, Ausbildung und Beruf, 4. Auflage*. Dudenverlag, 2006.

[Fel93] Michael R. Fellows. Computer science and mathematics in the elementary schools. In Harvey B. Keynes Naomi Fisher and Philip D. Wagreich, editors, *Mathematicians and Education Reform 1990–1991*, volume 3 of *CBMS Issues in Mathematics Education*. American Mathematical Society, 1993. `https://ianparberry.com/research/cseducation/fellows1991.pdf`.

[Kap14] Manu Kapur. Productive failure in learning math. *Cognitive Science*, 38(5):1008–1022, 2014. `https://doi.org/10.1111/cogs.12107`.

[Str23] Gilbert Strang. *Introduction to Linear Algebra (Sixth Edition)*. Wellesley-Cambridge Press, 2023.

[War01] William P. Wardlaw. A generalized general associative law. *Mathematics Magazine*, 74(3):230–233, 2001. `https://doi.org/10.1080/0025570X.2001.11953069`.

Index