Foundations of Mathematics 24/25
Linear Algebra

Alexander Koller, Nicolas Faroß, Michael Hahn

October 12, 2024

1 Introduction

These are some notes for the course “Foundations of Mathematics” / “Mathematische Grundlagen: Lineare Algebra und Analysis” at the Department of Language Science and Technology at Saarland University.

It is not intended as a standalone introduction to mathematics. Instead, it is a companion document to the main teaching materials (video or reading). Its purpose is to pad out some technical details and establish notational conventions.

The primary teaching materials we will use for the linear algebra section of this course are the Youtube videos by 3Blue1Brown, which are very accessible and intuitive, and the lecture notes by Maclagan and Testa from the University of Warwick (M&T). The lecture notes are a densely written technical text.

Learning how to read math texts such as M&T can be hard work, but it is a crucial skill to acquire if you want to read current NLP papers. You may need to read some parts multiple times and take some time to think and digest them. Make sure you understand and remember definitions in technical detail and that you understand each step of the proofs; and when M&T asks you to check for yourself that some statement is true, take a moment to think it through yourself. It is okay if you don’t understand everything; this is why we meet in class so I can explain. Just make sure to ask questions: if you don’t, I will assume that you understood the point and may ask you to explain it in class.

Notice that each section in this document is marked with the date on which we will discuss its content in class. Make sure you have prepared each section on time.

2 Vectors and Matrices — end of October

- ▶ 3Blue1Brown: Vectors, what even are they?
- ▶ M&T, Sections 1–2

3Blue1Brown suggests multiple perspectives on vectors. One way of thinking about them is as tuples of numbers, and he writes them as *column vectors:*

$$\begin{pmatrix}1\\2\end{pmatrix} \qquad \begin{pmatrix}0\\-3\\0.5\end{pmatrix} \qquad \begin{bmatrix}-2\\7\end{bmatrix}$$

Vectors and matrices are sometimes written with square brackets around them (e.g. 3B1B) and sometimes with round brackets (e.g. Maclagan and Testa's lecture notes). The difference is purely one of taste.

What is *not* just a matter of taste is the difference between a *column vector* and a *row vector*. In M&T's notation, we can think of a column vector as an $n \times 1$ matrix, i.e. a matrix with a single column that holds n values in its n rows. We can think of a row vector as a $1 \times n$ matrix, i.e. a matrix with a single row that holds n values in its n columns.

- Column vector: $\begin{pmatrix}1\\2\end{pmatrix}$
- Row vector: $\begin{pmatrix}1 & 2\end{pmatrix}$

The difference may seem purely cosmetic to you at this point, but it will become very important once we talk about how to multiply matrices with vectors. Throughout this document, I write all vectors and matrices in round brackets. As special notation, I will sometimes write $[1, 2]$ (notice the square brackets) in running text as a shorthand for the column vector shown above. This is my own private notational convention to keep the typography of the document neat; please don't use this notation outside of this class.

Variables for vectors are written in boldface, e.g. $\mathbf{a}, \mathbf{b}, \mathbf{c}$. Sometimes, variables for vectors instead have an arrow on top of them, $\vec{a}$; this is just a different notational convention for the same thing. The value of such a variable is always a vector, e.g.

$$\mathbf{a} = \begin{pmatrix}2\\3\end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix}4\\5\end{pmatrix}$$

We write the i-th element of the vector $\mathbf{a}$ as a_i, and similarly for other vector variables. Thus e.g. $b_2 = 5$.

Vector arithmetic. You can add vectors to each other and multiply vectors with scalars, i.e. with real numbers from $\mathbb{R}$. This is just a special case of the way matrices are added and multiplied in M&T. Observe the neat geometric intuitions behind addition and scalar multiplication.

In standard vector arithmetic, you *cannot* multiply one vector by another vector. You can only multiply a vector with a scalar, which makes the vector longer or shorter.

There is an operation called *element-wise multiplication* on two vectors, which is sometimes written $\mathbf{a} \otimes \mathbf{b}$ or $\mathbf{a} \odot \mathbf{b}$: If you have vectors $\mathbf{a}, \mathbf{b}$, you let $c_i = a_i \cdot b_i$ and take $\mathbf{c}$ as the product of $\mathbf{a}$ and $\mathbf{b}$. But this operation does not have a clean geometric intuition – check this –, and thus we never really look at this operation in linear algebra. However, this operation is used quite frequently in machine learning and NLP papers, so it is worth knowing. Usually the paper will say "elementwise multiplication" (and you definitely should say that in your own papers and MSc thesis), but if they don't explain what multiplication they mean, elementwise is usually a plausible candidate. Note that in pure math texts $\mathbf{a} \otimes \mathbf{b}$ usually denotes a different operation called tensor product.

There is an operation called the *dot product* or *inner product*, $\mathbf{a} \cdot \mathbf{b}$, which is very useful in linear algebra and has a nice geometric operation. However, the result of the dot product is not a vector, but a real number. We will talk about dot products in more detail in about a month.

Every vector space has a *zero vector*,

$$\mathbf{0} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

All coefficients of the zero vector are zero; the zero vector with n zeroes is called $\mathbf{0}_{n1}$ in M&T. Thus, $\mathbf{a} + \mathbf{0} = \mathbf{a}$ for all vectors $\mathbf{a}$. Note also that $a \cdot \mathbf{0} = \mathbf{0}$ for all scalars a.

Matrix arithmetic. In M&T, matrices are introduced. Similar to vectors, these can be added and multiplied by numbers. However, there is also a multiplication for matrices. At the moment, the multiplication formula might look strange but we will later see that it has a nice geometric interpretation.

In Proposition 2.7., some rules for working with matrices are stated. These are similar to the rules for ordinary numbers, but there are some differences one has to keep in mind. For example, it is in general not possible to divide by a matrix and it does not hold that $AB = BA$ for arbitrary matrices A and B. Also, one has to pay attention to the dimension of matrices, since AB is only defined if the number of columns of A is the same as the number of rows of B.

Vector spaces. For now, we assume that all scalars are real numbers (i.e., elements of $\mathbb{R}$), and that all vectors are tuples of real numbers. We write $\mathbb{R}^n$ for the vector space of n-tuples. We will define vector spaces more generally next time.

3 Solving linear equation systems

end of October

- ▶ The Game of Gaussian Elimination: An Introduction to Linear Algebra
- ▶ Math Hacks: Solve 3x3 systems with matrices (Gaussian elimination)
- ▶ M&T, Sections 3.1–3.5
- ▶ Practice: Random linear equation systems

Solving systems of linear equations (SLEs) is a problem that is very intricately tied to the field of linear algebra. You will solve many linear equation systems in the exercises that we will do in the next few weeks, so practicing the Gauss algorithm is a very good use of your time.

Brett from Math Hacks explains the Gauss algorithm for 3×3 equation systems in her blog post and video, i.e. for systems that consist of three linear equations with three variables. Obviously, nothing hinges on this particular choice of problem size, and M&T explain the method in full generality. Notice that there are subtle differences between the transformation rules that Brett and M&T use; in addition, Brett writes the row into which the result should be written to the right of the arrow, whereas M&T write it to the left of the arrow.

When you look at a system of linear equations, there are three possibilities:

1. The system has a *unique* solution (e.g. Example 2 in Section 3.2 of M&T).
2. The system is inconsistent, i.e. it has *no* solution (e.g. their Example 1).
3. The system is underconstrained, i.e. it has *infinitely many* solutions (e.g. their Examples 3 and 4).

You should have a solid understanding of the outcome of the Gauss algorithm in each of these three cases.

One could say that the whole point of linear algebra is to explain the difference between these three cases and to relate it to geometric concepts, in a way that is very satisfying and really beautiful.

Howto: Extracting solutions out of the upper echelon form. Once you have brought the augmented matrix for an SLE into upper echelon form, there is the question of how you get the set of solutions. You can proceed as follows.

Work your way through the augmented matrix in upper echelon form, row by row, from bottom to top. By design of the upper echelon form, there will be some new variables in each step that have a nonzero coefficient, but had zero coefficients in all lower rows. Let's call the leftmost of these variables the *leading variable* in this row. All other new variables in the row are *free variables*; you will find that you can choose their values freely, as long as you select appropriate values for the non-free variables.

In each row, express the leading variable in terms of the free variables you have encountered so far. Then in the end, to find your set of solutions, you let the values of the free variables range over all real numbers, and the other values will have their values determined by the free variables.

Consider the following example SLE.

$$
\begin{aligned}
x_1 + 2x_2 + x_3 &= 0 \\
3x_1 + 7x_2 + 2x_3 &= 0 \\
2x_1 + 5x_2 + x_3 &= 0
\end{aligned}
\tag{1}
$$

The augmented matrix for this SLE looks as follows:

$$
\begin{pmatrix}
1 & 2 & 1 & 0 \\
3 & 7 & 2 & 0 \\
2 & 5 & 1 & 0
\end{pmatrix}
\tag{2}
$$

By carrying out the Gauss algorithm, you can bring this matrix into the following upper echelon form:

$$
\begin{pmatrix}
1 & 2 & 1 & 0 \\
0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\tag{3}
$$

Now let's go through the rows one by one. The third row has no variables with nonzero coefficients, so we can skip it.

The second row has two variables that have nonzero coefficients, but had zero coefficients in all lower rows: x_2 and x_3. x_2 is the leading variable in the second row, and x_3 is therefore a free variable. The second row corresponds to the equation $x_2 - x_3 = 0$, so we can express the leading variable in terms of the free variables as

$$
x_2 \quad = \quad x_3. \tag{4}
$$

Finally, the first row introduces one new variable with nonzero coefficients, x_1. Thus x_1 is the leading variable of the first row; there are no new free variables in addition to x_3, which we already collected as a free variable. The first row expresses the equation $x_1 + 2x_2 + x_3 = 0$, which we can solve for the leading variable as follows:

$$
\begin{aligned}
x_1 &= -2x_2 - x_3 \\
&= -3x_3
\end{aligned}
\tag{5}
$$

Thus, we have expressed all non-free variables (x_1, x_2) in terms of the free variables (x_3). We can choose a value for x_3 freely, and then the equations (4) and (5) tell us how to construct values for x_1 and x_2 out of the value for x_3 that will solve the SLE. This means that the solution set of the SLE looks as follows:

$$
\{(-3x_3, x_3, x_3) \mid x_3 \in \mathbb{R}\} \tag{6}
$$

Tips for solving SLEs. The most important tip for solving SLEs is to *work carefully and systematically.* It is very easy to confuse yourself: what rows have I already transferred to the new matrix? What exact operation am I performing to generate the row I'm currently working on? If you forget whether the new row is $r_1 - r_2$ or $r_2 - r_1$, you will make mistakes and obtain an incorrect solution.

Similarly, double-check all your calculations. It is very easy to make mistakes in subtracting negative numbers; $1 - (-5)$ is 4, not -6.

When you're done, you can check your solution by substituting it into the original linear equations and testing if it actually solves all the equations.

Above all, carrying out the Gauss algorithm is an acquired skill. There is no substitute for practice. I cannot overstate the usefulness of solving lots of SLEs at home between classes, until you can do it reliably and without much thought.

4 Questions to think about

end of October

Here are some questions that I'd like you to think about before class. There will be similar questions for the other classes. I'd like to discuss the questions with you in class, so the more carefully you prepare, the better!

(a) So, what even is a vector?

(b) The lecture notes use technical mathematical language, such as the symbol "$\in$" for *being an element of a set*, "$\subseteq$" for *subset* or the word *axiom*. Which symbols, words and phrases are unknown to you, or seem to be used in an unfamiliar way?

(c) What differences can you find between the lecture notes and the videos (besides of course the text being more technical)? Are these just differences in notation, or are they more substantial?

5 In-class activities

end of October

Here are some activities that we will work on in class.

You should feel free to try these activities by yourself at home. This will give you a sense of whether you understood the concepts for this class, and help you come up with questions that we can discuss together.

(a) Come up with a vector in $\mathbb{R}^2$ (i.e. pairs of numbers) and ask your partner(s) to draw them as arrows in a coordinate system.

(b) Come up with arrows in a coordinate system and ask your partner(s) to represent them as tuples of numbers.

(c) Come up with a few vectors as above, and ask your partner(s) to add and subtract them and multiply them with scalars. Do this both in the numbers view and in the arrows view.

(d) Using the random generator linked above, solve a few linear equation systems and have your partner(s) check your work.

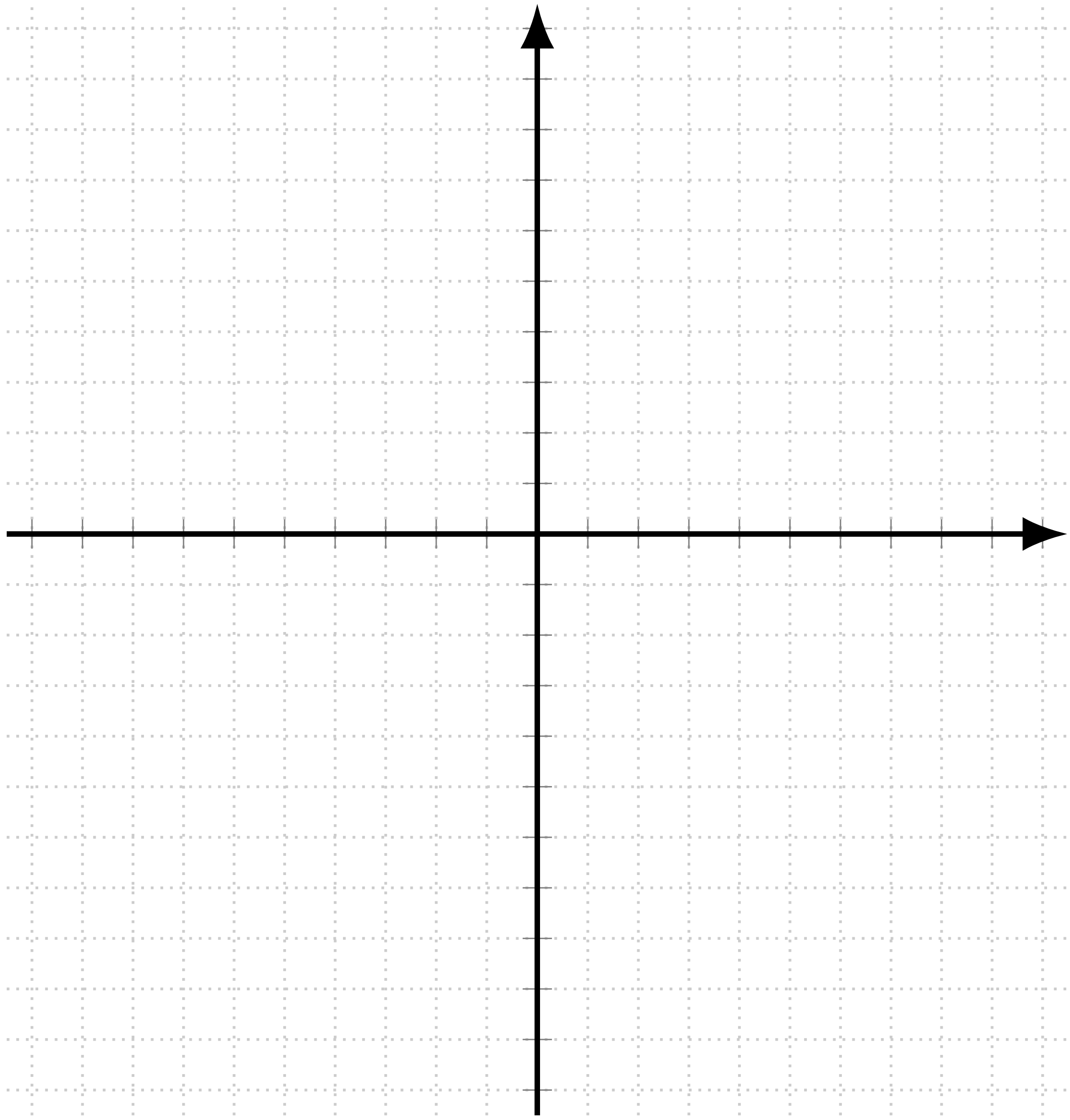
(e) Come up with a few matrices and ask your partner(s) to multiply them. Can you find square matrices A, B such that $AB \neq BA$?

(f) Convince yourself that (i)–(iv) in M&T, Proposition 2.7. hold. If you like you can try to prove them.

6 Coordinate system to draw onto

end of October

Here is a coordinate system that you can use for the in-class exercise.



7 Vector spaces

end of October

- M&T, Sections 4–5
- 3Blue1Brown: Abstract vector spaces

One fundamental difference between 3B1B and M&T is that 3B1B leans very heavily on geometric intuitions about linear algebra, whereas M&T take a more abstract perspective. While 3B1B assumes throughout his videos that we work with the vector space $\mathbb{R}^n$ of n-tuples of real numbers, M&T give the abstract definition of vector spaces that mathematicians use.

What makes this abstract definition so useful is it can also be applied to a large variety of other examples, e.g. matrices, polynomials, functions and even some finite sets. For the moment it will be enough to think of vector spaces just as $\mathbb{R}^n$, but it might be good to keep those other examples in mind. If we have enough time, we will come back to them and see some nice applications of the concepts we will learn in the next weeks.

As a mathematician, 3B1B agrees that the abstract view on vector spaces is extremely useful and important; it's just that as a teacher, he thinks that it is better to build the geometric intuitions first and then generalize to the abstract view later. This is why his video on abstract vector spaces comes at the very end of his Linear Algebra playlist. As a consequence, he talks about many concepts from linear algebra that you are not yet familiar with. I recommend that you watch the video now, because it contains many great illustrations of the definitions in M&T, and just accept that there are certain things you can't understand yet. Then as you learn more, go and periodically revisit the video; your understanding will deepen every time. In particular, you should watch the video again after we have learned about linear transformations and matrices.

There is an important difference between the way in which 3B1B and M&T define vector spaces. 3B1B says that vectors can be anything that can be added together and multiplied *with a real number.* Arrows, tuples of numbers, and polynomials with real coefficients all have this property. M&T go one step further and say that vectors can be anything that can be added together and multipled *with a scalar from an arbitrary field.* This gives them the full generality at which a mathematician would teach this class, but it is overkill for our course. It is perfectly okay if you silently think "$\mathbb{R}$" whenever M&T talk about a field K. That said, I think the mathematicians' abstract perspective on fields and vector spaces is really beautiful, and I love the fact that you can define vector spaces over the craziest fields and all of linear algebra just works.

The most important thing you should take away for now is the concept of a *vector space*, as a set of values (called "vectors") where you can add vectors and multiply vectors with scalars from the underlying field. The most well-known vector spaces are the $\mathbb{R}^n$ spaces, and we will continue to focus on them here. But nothing in linear algebra hinges on any particular properties of $\mathbb{R}^n$; it works for arbitrary vector spaces.

8 Linear combinations and spans

end of October

- 3Blue1Brown: Linear combinations, spans, and basis vectors
- M&T, Section 6.1 & 6.2

Spans. The set

$$\mathsf{span}(\mathbf{c}, \mathbf{d}) = \{a\mathbf{c} + b\mathbf{d} \mid a, b \in \mathbb{R}\}$$

is called the *span* of $\mathbf{c}$ and $\mathbf{d}$. It is the set of all linear combinations of $\mathbf{c}$ and $\mathbf{d}$. Note that we sometimes leave the multiplication dot away for brevity: $a\mathbf{c}$ means the same thing as $a \cdot \mathbf{c}$. Note also that when you multiply a scalar with a vector, you always write the scalar to the left and the vector to the right,

In the video, 3B1B hints that the definition of span can be generalized to the span of arbitrarily many vectors. To write this out explicitly, say that $S \subseteq V$ is a set of vectors from the vector space V. Then we define the span of S as

$$\mathsf{span}(S) = \{a_1\mathbf{v}_1 + \ldots + a_n\mathbf{v}_n \mid a_1, \ldots, a_n \in \mathbb{R}, \mathbf{v}_1, \ldots, \mathbf{v}_n \in S\}.$$

Notice that S can be an infinite set. Linear combinations are only defined as weighted sums with finitely many summands, which is why the general definition allows us to select an arbitrary finite number of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ from S every time we generate a new element of the span. This is also why the definitions of linear dependence and spanning in M&T seem a bit indirect: They allow for infinite sets of vectors to be linearly (in)dependent and to span the vector space. This is important if we want our definitions to work for vector spaces of infinite dimension (e.g. the polynomials over $\mathbb{R}$), where no finite set of vectors will span the whole space.

Notice further that spans are also well-defined if S contains just one vector, or none at all. In the latter case, $\mathsf{span}(\emptyset) = \{\mathbf{0}\}$, because the sum of zero vectors is $\mathbf{0}$.[1]

Is there a linear combination? You can use the Gauss algorithm to find out whether a vector can be expressed as a linear combination of some other vectors. Let's say that $S = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\} \subseteq \mathbb{R}^n$ is a set of vectors, and we want to know if we can represent the vector $\mathbf{b} \in \mathbb{R}^n$ as a linear combination of the vectors in S. That is, we want to know whether there are scalars $a_1, \ldots, a_k \in \mathbb{R}$ such that

$$a_1\mathbf{v}_1 + \ldots + a_k\mathbf{v}_k = \mathbf{b}. \tag{7}$$

Each of the vectors $\mathbf{v}_i$ is an n-tuple of numbers. Let's say that $\mathbf{v}_i = [v_{i1}, \ldots, v_{in}]$; so we have real numbers $v_{ir} \in \mathbb{R}$ for every index $1 \leq i \leq k$ and every index $1 \leq r \leq n$. Then solving the equation (7) for the variables $a_1, \ldots, a_k$ amounts to solving the following system of n linear equations:

[1] Thanks to Niyati Bafna for pointing out a mistake in an earlier version of these notes.

$$
\begin{aligned}
v_{11}a_1 + \ldots + v_{1k}a_k &= b_1 \\
\ldots & \quad \ldots \\
v_{n1}a_1 + \ldots + v_{nk}a_k &= b_n
\end{aligned} \tag{8}
$$

This SLE has a solution $(a_1, \ldots, a_k)$ iff[2] this tuple of a's also solves the vector equation (7). Indeed, you could say that the vector equation is simply a compact way of writing down exactly the same thing as the SLE in (8).

Are the vectors linearly independent? You can use the same trick to find out whether a set S of vectors is linearly independent. M&T do this in Example 6.3. They spell out their vector equation $\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 = \mathbf{0}$ into the SLE $\alpha_1 + 2\alpha_2 = 0$ and $3\alpha_1 + 5\alpha_2 = 0$. Using the Gauss algorithm, you can find out that there is a unique solution $(0, 0)$ (do that). As a consequence, there is no nontrivial solution of the equation. The only way by which $\mathbf{0}$ can be expressed as a linear combination of $\mathbf{v}_1$ and $\mathbf{v}_2$ is by multiplying both vectors by zero, meaning that $\mathbf{v}_1$ and $\mathbf{v}_2$ are linearly independent.

By contrast, their Example 6.4 generates an SLE that has nontrivial solutions, so those vectors are linearly dependent.

Note that Lemma 6.5 offers another method for testing whether a set S of vectors is linearly dependent. First, you check if S contains the zero vector; if yes, then S is linearly dependent, because every set that contains the zero vector is linearly dependent. Second, you check if one of the vectors in S can be expressed as a linear combination of the others. We saw how to do that with SLEs above. Note that in this case, we do not require that the linear combination is nontrivial; any linear combination will establish linear dependence.

Howto: Checking for linear dependence. Here are some methods for checking a set S of vectors from a vector space V with dimension n for linear dependence.

1. If one of the vectors is $\mathbf{0}$, then S is linearly dependent.
2. If S contains only two vectors, then it is linearly dependent if and only if one of the vectors is a multiple of the other. This is usually obvious by visual inspection.
3. Express $\mathbf{0}$ as a linear combination of S and solve the corresponding SLE; S is linearly dependent iff the SLE has a nontrivial solution. (This is the case if it has infinitely many solutions, because $\mathbf{0}$ will always be a solution of this SLE.)
4. Pick one vector from S and express it as a linear combination of the other vectors. S is linearly dependent if the corresponding SLE has a solution (including one where all coefficients are zero).

[2] This is not a typo. In mathematical texts, "iff" is an abbreviation for "if and only if". Check that "if" and "only if" is not the same thing.

Do the vectors span the space? If you want to check whether a set S of vectors spans a vector space V, you can do this by letting $\mathbf{b}$ be an *arbitrary* vector of V. That is, you simply make a linear equation system as in (8), and you insert the actual coefficients of the vectors in S on the left-hand side of the equations. But instead of using actual numbers for $b_1, \dots, b_n$, you simply leave those as $b_1, \dots, b_n$. Thus if you can solve the SLE, you will obtain a recipe for selecting the $a_1, \dots, a_k$ given any particular $\mathbf{b}$ that you want.

Let me explain what I mean with an example. Let's say you want to know if the set $S = \{\mathbf{v}_1, \mathbf{v}_2\}$ with $\mathbf{v}_1 = [1, 0]$ and $\mathbf{v}_2 = [0, -1]$ spans the vector space $\mathbb{R}^2$. Let $\mathbf{b} = [b_1, b_2]$ be an arbitrary vector in $\mathbb{R}^2$. Then the SLE looks as follows:

$$\begin{aligned} 1 \cdot a_1 + 0 \cdot a_2 &= b_1 \\ 0 \cdot a_1 + (-1) \cdot a_2 &= b_2 \end{aligned} \tag{9}$$

If you apply the Gauss algorithm, you obtain the following row-reduced form of the augmented matrix:

$$\begin{pmatrix} 1 & 0 & b_1 \\ 0 & 1 & -b_2 \end{pmatrix} \tag{10}$$

That is, no matter what value you pick for b_1 and b_2, you know that $a_1 = b_1$ and $a_2 = -b_2$ is the (unique) solution of this SLE. Thus you know that you can express every vector $\mathbf{b} \in \mathbb{R}^2$ as a linear combination of S, meaning that S spans the entire vector space.

Obviously, if the SLE is inconsistent, at least for some values of $\mathbf{b}$, then you know that S does *not* span the space. For example, if you have $\mathbf{v}_1 = [1, 0]$ and $\mathbf{v}_2 = [2, 0]$, then you obtain this SLE:

$$\begin{aligned} 1a_1 + 2a_2 &= b_1 \\ 0a_1 + 0a_2 &= b_2 \end{aligned} \tag{11}$$

Simply by looking at this SLE, you can see that it can only be solved if $b_2 = 0$. In this case, any choice of (a_1, a_2) will satisfy the second equation; but if $b_2 \neq 0$, then there is obviously no choice of (a_1, a_2) that will satisfy it. Thus, not every vector of $\mathbb{R}^2$ can be expressed as a linear combination of S. One counterexample is $[0, 1]$, because $b_2 = 1 \neq 0$.

9 Questions to think about

end of October

(a) What exactly are the "if" and the "only if" directions of the statement that (7) has a solution iff (8) has a solution? Explain why both statements are true.

(b) Above I said that every set of vectors that contains the zero vector is linearly dependent. Why is this true?

(c) Rewrite the condition of Definition 6.6 of M&T using the concept of "span" defined above.

(d) Convince yourself that the statements (i) to (iv) at the end of Section 5 in M&T hold. You can first think of the special case where $K = \mathbb{R}$ and $V = \mathbb{R}^n$ for some n; then think about the general case of arbitrary fields K and vector spaces V.

10 In-class activities

end of October

(a) Pick a set of vectors $\mathbf{v_1}, ..., \mathbf{v_n}$ and an additional vector $\mathbf{b}$ and ask your partner to test if $\mathbf{b}$ can be written as linear combination of $\mathbf{v_1}, ..., \mathbf{v_n}$.

(b) Pick a set of vectors and ask your partner to test them for linear dependence. Try both algorithms: by expressing $\mathbf{0}$ as a nontrivial linear combination, or by expressing one vector as a linear combination of the others.

11 Bases and dimensions

≈ November

- 3Blue1Brown: Linear combinations, spans, and basis vectors
- M&T, Section 6.3 & 6.4

Bases. If V is a vector space, it sometimes happens that $V = \mathsf{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$, i.e. every vector in V can be expressed as a linear combination of the set $B = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$. In this case, we say that V is *spanned by* B. For example, $[0,1]$ and $[1,0]$ span the vector space $\mathbb{R}^2$.[3]

If V is spanned by the vectors in B, and if the vectors in B are also linearly independent, we call B a *basis* of V. Bases are really important because every vector in V can be expressed as a linear combination of B in exactly one way.

For example,

$$B_1 = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$

is a basis of $\mathbb{R}^2$: An arbitrary vector $[a,b]$ can be written as the linear combination $a \cdot [0,1] + b \cdot [1,0]$, but B_1 itself is linearly independent.

The set

$$B_2 = \left\{ \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}$$

is another basis of $\mathbb{R}^2$ because $[a,b] = (-a) \cdot [-1,0] + (-b) \cdot [0,-1]$. Because B_2 is linearly independent, it is also a basis. In general, there are infinitely many bases of each vector space $\mathbb{R}^n$.

The set

$$B_3 = \left\{ \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 4 \end{pmatrix} \right\}$$

is *not* a basis of $\mathbb{R}^2$. For instance, the vector $[0,1]$ cannot be expressed as a linear combination of B_3 (try it); furthermore, B_3 is linearly dependent.

Dimensions. It can be shown that for every vector space V, every basis of V contains the same number of vectors. This number is called the *dimension* of V. Another way of defining the dimension is the maximum number of linearly independent vectors you can find in V. The idea is that if you play a game in which you start with a single vector (not $\mathbf{0}$) and then add another linearly independent vector in each round, you will eventually run out of choices for linearly independent vectors because your set spans the entire vector space.[4]

[3] Remember that $[0,1]$ is only my private notation for a column vector.

[4] This will only work if the dimension is a finite number. There are also infinite-dimensional vector spaces. Then the technical definitions of "dimension" still hold up, but your game would never end.

In the examples above, you saw that B_1 and B_2 both have exactly two vectors in them. This means that the dimension of $\mathbb{R}^2$ is two. Thus if you ever see a set of three or more vectors from $\mathbb{R}^2$, you know that they must be linearly dependent.

The set B_3 also consisted of two vectors, but they were not a basis of $\mathbb{R}^2$. This is because they are linearly dependent. Therefore the dimension of $\mathsf{span}(B_3)$ is one. A basis of this span is $\{[1, 2]\}$.

Howto: Constructing a basis. To construct a basis from a set S, you first have to check that it is linearly independent. If it is not linearly independent, you can sift it to remove the linearly dependent vectors (see M&T, Example 6.14).

Second, check whether the vectors in S span V. The default method is to check that all vectors in V can be expressed as a linear combination of S, see Section 8. As a shortcut, Corollary 6.19 in M&T says that if S contains n linearly independent vectors, then it spans $\mathbb{R}^n$.

If S does *not* span V, successively add further linearly independent vectors until you have n vectors. One way to do it is to check which vectors $\hat{\mathbf{\imath}}, \hat{\mathbf{\jmath}}, \ldots$ from the standard basis are not in the span and use those. Alternatively, you can make small changes to vectors in S (e.g. add one to a coefficient) and double-check that they are not in the span.

Howto: Tricks for checking linear independence/spanning. When we want to check if some vectors $\mathbf{v_1}, \ldots, \mathbf{v_m} \in V$ are linearly independent/spanning and we know that V has dimension n then Corollary 6.19 and Corollary 6.20 allow us sometimes to use the following shortcuts:

1. If $m < n$ then $\mathbf{v_1}, \ldots, \mathbf{v_m}$ are not spanning.
2. If $m > n$ then $\mathbf{v_1}, \ldots, \mathbf{v_m}$ are not linearly independent.
3. If $m = n$ then $\mathbf{v_1}, \ldots, \mathbf{v_m}$ are linearly independent if and only if $\mathbf{v_1}, \ldots, \mathbf{v_m}$ are spanning.

Why bases? At the moment it might not be clear why it should be useful to consider a different basis than $[1, 0, 0, \ldots], [0, 1, 0, \ldots], \ldots$ for $\mathbb{R}^n$. For this, we have to wait until we introduce linear maps and see how these are related to matrices.

However, if we work with more abstract vector spaces, bases are already really powerful. Consider for example the vector space

$$V = \{f \mid f \text{ is a quadratic polynomial}\}.$$

In this description, V might look complicated. But if we use that fact that 1, x, x^2 is a basis of V, we can write it as

$$V = \{ax^2 + bx + c \mid a, b, c \in \mathbb{R}\} \approx \{[a, b, c] \mid a, b, c \in \mathbb{R}\} = \mathbb{R}^3.$$

Hence, each quadratic polynomial is uniquely described by three real numbers a, b, c and the space V looks like $\mathbb{R}^3$, which is more familiar. Similarly, when we have any abstract vector space V with dimension n, we can use a basis to introduce coordinates and view it as $\mathbb{R}^n$:

$$\begin{array}{ccc} V \ + \ \text{basis } \mathbf{v_1}, \ldots, \mathbf{v_n} & \approx & \mathbb{R}^n \\ \alpha_1 \mathbf{v_1} + \ldots + \alpha_n \mathbf{v_n} & \longleftrightarrow & [\alpha_1, \ldots, \alpha_n] \end{array}$$

This also justifies why it is enough for us to focus only on $\mathbb{R}^n$.

12 Subspaces

$\approx$ **November**

▶ M&T, Section 7

Subspaces. Subspaces are subsets of a vector space which are again vector spaces. Geometrically, subspaces of $\mathbb{R}^3$ are lines and planes containing the origin. Further, there is exactly one zero-dimensional subspace containing only the origin and one 3-dimensional subspace containing all of $\mathbb{R}^3$.

Defining subspaces. One way to define a subspace is by writing down a set of vectors which span it (see Proposition 7.8.).

Alternatively, subspaces are often defined via a property of vectors. For example, consider $V = \mathbb{R}^3$ and the subset

$$W = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \mid x = y \right\}.$$

To see that W is a subspace, we have to check that it is closed under addition and scalar multiplication:

1. If $\mathbf{w_1} = [x_1, y_1, z_1], \mathbf{w_2} = [x_2, y_2, z_2] \in W$ then $\mathbf{w_1} + \mathbf{w_2} \in W$, because

$$\mathbf{w_1} + \mathbf{w_2} = [x_1 + x_2, y_1 + y_2, z_1 + z_2]$$

and

$$(x_1 + x_2) - (y_1 + y_2) = (x_1 - y_1) + (x_2 - y_2) = 0 + 0 = 0.$$

2. If $\mathbf{w} = [x, y, z] \in W$ and $\alpha \in \mathbb{R}$ then $\alpha\mathbf{w} = [\alpha x, \alpha y, \alpha z] \in W$, because

$$\alpha x - \alpha y = \alpha(x - y) = \alpha 0 = 0.$$

Hence, W is indeed a subspace of $\mathbb{R}^3$.

13 Questions to think about

$\approx$ **November**

(a) What is the dimension of $\mathbb{R}^2$? Of $\mathbb{R}^3$? Can you see a pattern that generalizes to $\mathbb{R}^k$ for arbitrary k, and why is it true? What happens with $k = 1$ and $k = 0$?

(b) Does the technical definition of *dimension* given in the lecture notes fit with how you think about dimensions in real life (e.g. 3D vs 2D movies)?

(c) What are some examples of subsets of a vector space which are not subspaces?

(d) Consider a span of two arbitrary vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$. Are $\mathbf{v}_1$ and $\mathbf{v}_2$ together always a basis of the span they create?

(e) Prove Proposition 7.8 in M&T by checking the definition of subspace.

14 In-class activities

$\approx$ **November**

(a) Pick a set of vectors and ask your partner to check whether they are a basis of $\mathbb{R}^2$. If they are not, fix them as follows:

- If they are linearly dependent, sift them until they are a basis.
- If they do not span the vector space, complete them into a basis by adding more vectors. Feel free to use geometric intuitions where they are helpful. Note also that by definition, a vector that is not in the span of S is linearly independent of all other vectors in S.

Try vector sets of different sizes (i.e. with 0, 1, 2, 3 vectors in them). Once you get the hang of $\mathbb{R}^2$, try the same thing for $\mathbb{R}^3$ too.

(b) Consider the subspace $W = \mathsf{span}(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4) \subseteq \mathbb{R}^3$, where

$$\mathbf{w}_1 = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}, \qquad \mathbf{w}_2 = \begin{pmatrix} -3 \\ 3 \\ 3 \end{pmatrix}, \qquad \mathbf{w}_3 = \begin{pmatrix} 2 \\ -2 \\ 4 \end{pmatrix}, \qquad \mathbf{w}_4 = \begin{pmatrix} -1 \\ 1 \\ 3 \end{pmatrix}.$$

- Why do the vectors $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4$ not form a basis of W?
- Use the sifting strategy from Example 6.14. to compute a basis of W. What is the dimension of W?
- Sketch W in a coordinate system.
- Extend the basis of W to a basis of $\mathbb{R}^3$. How many vectors do you have to add?

15 Linear transformations and matrices

≈ **November**

- 3Blue1Brown: Linear transformations and matrices
- 3Blue1Brown: Matrix multiplication as composition
- 3Blue1Brown: Three-dimensional linear transformations
- M&T, Section 8

In this section, we come to a core idea of linear algebra: that you can map one vector space into another using a *linear transformation*, and that you can represent this linear transformation with a *matrix*. Once again, 3B1B takes an almost entirely geometric perspective, and M&T take a more formal perspective. The beauty of it is that both perspectives are equivalent (on $\mathbb{R}^2$ and $\mathbb{R}^3$, because we don't really have good geometric intuitions about more complicated vector spaces), and it is really useful and healthy to go back and forth between both perspectives.

I recommend that you go through Example 8.3 in M&T very carefully. If anything is unclear, let's talk about it in class. These examples will help you come to a more complete appreciation of what you can do with linear transformations, beyond what 3B1B visualizes in the videos. **Note:** There are typos in the first example in M&T; the correct matrices for $T(\alpha, \beta) = (2\alpha + \beta, 3\alpha - \beta)$ (as in Example 8.3) and $T(\alpha, \beta) = (3\alpha + \beta, 2\alpha - \beta)$ (as in Example 8.9) are

$$\begin{pmatrix} 2 & 1 \\ 3 & -1 \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} 3 & 1 \\ 2 & -1 \end{pmatrix},$$

respectively.

Observe that M&T write $K^{m,n}$ for the set of all matrices with m rows and n columns with coefficients in the field K. 3B1B shows only matrices from $\mathbb{R}^{2,2}$ and $\mathbb{R}^{3,3}$ in his videos. Notice that you can choose m and n freely, and in particular they don't have to be the same; this is useful e.g. if you want to represent a linear transformation $T : U \to V$ with a matrix where the dimensions $n = \dim(U)$ and $m = \dim(V)$ are different.

One concept that you've probably struggled with when reading M&T is that of a dual space (end of 8.2). Don't worry about this for now; we will discuss it in more detail later.

Coordinates of vectors. A matrix uniquely represents a linear transformation $T : U \to V$, *with respect to bases E* of U and F of V. If you choose a different basis, you get a different matrix for T. To see this, it is useful to understand vectors in terms of *coordinates*. M&T and 3B1B both distinguish between vectors and their coordinates, but I think, not clearly enough, so let's introduce some notation.

If we fix a basis $E = \{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ of a vector space V, we can represent each vector $\mathbf{v} \in V$ by its coordinates with respect to E. As you know, you can write $\mathbf{v}$ as a linear combination of E,

$$\mathbf{v} = a_1\mathbf{e}_1 + \ldots + a_n\mathbf{e}_n,$$

with a unique tuple of scalars $a_1, \ldots, a_n \in \mathbb{R}$. We can call the tuple $\langle a_1, \ldots, a_n \rangle$ the *coordinates* of $\mathbf{v}$ with respect to E. Given a coordinate tuple $\underline{\mathbf{a}} = \langle a_1, \ldots, a_n \rangle$ and a basis E, we can *evaluate* it to a vector

$$\mathbf{v} = \underline{\mathbf{a}}_E = a_1\mathbf{e}_1 + \ldots + a_n\mathbf{e_n}.$$

Note that we follow M&T in using underlined variables $\underline{\mathbf{v}}$ for coordinate tuples. But M&T call these "vectors", and I think this is misleading; they are tuples of scalars, disembodied from the actual vector space. (Of course, in $\mathbb{R}^n$, the vectors *are* tuples of scalars. But this is a special case; in most vector spaces, the vectors are not tuples of scalars. Also, if E is not the standard basis, even in $\mathbb{R}^n$ the coordinates will not be the same as the coefficients in the vector.)

Note further that the coordinates of the basis vectors themselves are really simple. The coordinates of $\mathbf{e}_1$ are $\langle 1, 0, \ldots, 0 \rangle$; the coordinates of $\mathbf{e}_2$ are $\langle 0, 1, 0, \ldots, 0 \rangle$; and so on.

Howto: Representing linear transformations as matrices. Now let $T : U \to V$ be a linear transformation, and let $\mathbf{u} \in U, \mathbf{v} \in V$ be vectors such that $\mathbf{v} = T(\mathbf{u})$. The matrix A of T with respect to the bases E and F describes how the coordinates of $\mathbf{u}$ with respect to E are transformed into the coordinates of $\mathbf{v}$ with respect to F.

In our language, the i-th column vector of A consists of the coordinates of $T(\mathbf{e}_i)$ with respect to the basis $F = \{\mathbf{f}_1, \ldots, \mathbf{f}_m\}$ of V. Thus the choice of basis of U determines which vectors the columns are based on, and the choice of basis of V determines the coordinate system in which the images $T(\mathbf{e}_i)$ are expressed.

You can think of the matrix encoding the linear transformation as follows, based on fixed bases E of U and F of V.

1. In order to map a vector $\mathbf{u} \in U$, first encode it as the coordinate tuple $\underline{\mathbf{u}}$ with respect to E.
2. Multiply A with $\underline{\mathbf{u}}$. This will produce another coordinate tuple $\underline{\mathbf{v}}$.
3. Decode $\underline{\mathbf{v}}$ into a vector $\mathbf{v} \in V$, with respect to F. Then you will have $\mathbf{v} = T(\mathbf{u})$.

Therefore, if you want to compute the matrix of a linear transformation, you first determine $T(\mathbf{e}_1), \ldots, T(\mathbf{e}_n) \in V$. Then you determine the coordinates of each of these vectors with respect to F. This will give you the columns of A.

Linear equations and linear transformations. Linear transformations and systems of linear equations are very closely connected.

Consider the following SLE, which has a unique solution $\mathbf{a} = [1, 2, -3]$.

$$\begin{array}{rcl} -3x_1 + 2x_2 + 4x_3 & = & -11 \\ -x_1 + x_3 & = & -4 \\ -x_2 & = & -2 \end{array} \tag{12}$$

We can rephrase this SLE as the problem of asking whether we can find a vector $\mathbf{x}$ which, if multiplied with a certain matrix A, will be mapped to a certain vector $\mathbf{b}$.[5] A consists of all the coefficients of the x_1, x_2, x_3 – one row per equation, one column per variable –, and $\mathbf{b}$ is the vector of results on the right-hand side:

$$A = \begin{pmatrix} -3 & 2 & 4 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} -11 \\ -4 \\ -2 \end{pmatrix} \tag{13}$$

If we let $\mathbf{x} = [x_1, x_2, x_3]$, then solving (12) is exactly the same as solving $A\mathbf{x} = \mathbf{b}$. In our example, letting $\mathbf{x} = \mathbf{a}$ will solve this matrix-vector equation.

Rewriting an SLE in this way has a number of advantages. First, the matrix-vector equation can be written much more compactly, which can be clearer to the reader because there is less visual noise. Second, by M&T's Proposition 8.10, the matrix A represents a linear transformation. We can thus access a geometric intuition on SLEs: Under the linear transformation encoded by A, what vector $\mathbf{x}$ will be mapped onto $\mathbf{b}$? This geometric intuition is visualized very neatly in Zach Star's video, linked above, and may help you get a new picture of why some SLEs have a unique solution and others have infinitely many. We will build upon this intuition later in this course.

Size constraints in matrix multiplications. Notice that Theorem 8.12 tacitly imposes constraints on the sizes of the matrices that can be multiplied. It assumes that T_2 maps vectors from U into vectors from V, and T_1 maps vectors from *that same* V into vectors from W; otherwise you couldn't apply T_1 to the outputs of T_2. This means in particular that you can only multiply two matrices if they share a dimension: AB is only defined if A is an $l \times m$-matrix and B is an $m \times n$-matrix, for some values of l, m, n. Thus, the number of columns of the left matrix (A) and the number of rows of the right matrix (B) *must be the same*; otherwise AB is undefined. The resulting matrix AB is then a $l \times n$-matrix, i.e. it has as many rows as A and as many columns as B. M&T explained this in the very beginning, in Definition 2.5.

Burn this relationship into your memory, because having size mismatches between matrices is one of the most common source of mistakes when you write neural network code. In my own code, I always add a comment to every single matrix I compute to record the size of that matrix, so I don't get confused later.

Column and row vectors. As mentioned in Section 2, row vectors (in tuple notation) can be seen as $1 \times n$-matrices, and column vectors can be seen as $n \times 1$-matrices. This means that if you have an $m \times n$-matrix $A \in \mathbb{R}^{m,n}$ representing the linear transformation T, and you have a *column* vector $\mathbf{u} \in \mathbb{R}^n$, you can compute $T(\mathbf{u})$ by treating $\underline{\mathbf{u}}$ as an $n \times 1$-matrix and simply multiplying A with $\underline{\mathbf{u}}$. Notice that A has n columns and $\underline{\mathbf{u}}$ has n

[5] Strictly speaking, I argued above that matrices do not transform vectors, but their coordinates. Let's assume here that we are in $\mathbb{R}^n$ with the standard basis so we don't have to make this distinction.

rows, so we are allowed to perform matrix multiplication. The result is an $m \times 1$-matrix, i.e. a column vector in $\mathbb{R}^m$.

By contrast, you can read a row vector $\mathbf{v}$ as an $1 \times m$-matrix and multiply it with A *from the left*: $\mathbf{v}A$ is a well-defined matrix multiplication and yields an $1 \times n$-matrix, i.e. a row vector in $\mathbb{R}^n$. However, it looks more natural to have the matrix (the "function") on the left, which is why we usually think more of column vectors than of row vectors.

Note that, again, strictly speaking, the preceding two paragraphs should talk about *coordinates* rather than vectors: column and row "vectors" and the columns of a matrix are coordinates of vectors. But nobody talks about "column coordinates", so that would just be confusing in a different way. Just be aware that whenever we we talk about matrix-matrix or matrix-"vector" multiplication, we are manipulating scalars with the addition and multiplication operations of the field, not the vector space. Thus everything is coordinates, not vectors.

Matrices in abstract vector spaces. While 3B1B works only with linear transformations within $\mathbb{R}^2$ and $\mathbb{R}^3$, Examples 8.3 and 8.9 in M&T discuss a number of more abstract examples. These require a bit more thought, but they illustrate just how powerful the idea of representing linear transformations as matrices is.

For instance, item 7 in Example 8.3 defines the "shift" operation $S_\alpha(f(x)) = f(x-a)$, where f is a polynomial in x. Let's assume that $K = \mathbb{R}$; then $f \in \mathbb{R}[x]_{\leq n}$, the $\mathbb{R}$-vector space of polynomials with real coefficients and rank up to n. The shift operation moves the graph of the polynomial to the right along the x-axis by α. So for instance, we have

$$S_3(x^2+2x-2) = (x-3)^2 + 2(x-3) - 2 = x^2 - 4x + 1 \in \mathbb{R}[x]_{\leq n}. \tag{14}$$

$\mathbb{R}[x]_{\leq n}$ is an $(n+1)$-dimensional vector space with the basis $B = \{1, x, \ldots, x^n\}$. Let's say for simplicity that $n = 2$. Then because B is a basis, all vectors (= polynomials) can be written as unique linear combinations $a \cdot 1 + b \cdot x + c \cdot x^2$: a, b, and c are simply the coefficients of the polynomial. Thus we can represent this polynomial as a column vector $[a, b, c]$. For instance, the polynomial $x^2 + 2x - 2$ from (14) becomes

$$\begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix}.$$

Now we encode S_3 as a matrix, just as explained in the 3B1B videos: by considering where it takes each basis vector. Each basis vector is mapped to a polynomial, and we can again represent each of these polynomials as a column vector with respect to the basis B:

$$\begin{array}{rcl} S_3(1) & = & 1 \\ S_3(x) & = & x-3 \\ S_3(x^2) & = & (x-3)^2 = x^2 - 6x + 9 \end{array} \qquad \begin{array}{rcl} \underline{1} & = & \langle 1, 0, 0 \rangle \\ \underline{x-3} & = & \langle -3, 1, 0 \rangle \\ \underline{x^2 - 6x + 9} & = & \langle 9, -6, 1 \rangle \end{array}$$

Thus the following matrix represents S_3 with respect to the basis B; its columns are the coordinates of the images of the basis vectors, from left to right. It is the special case of the matrix in Example 8.9 for $n = 2$ and $\alpha = 3$.

$$A = \begin{pmatrix} 1 & -3 & 9 \\ 0 & 1 & -6 \\ 0 & 0 & 1 \end{pmatrix}$$

We can double-check that this matrix actually captures S_3 by multiplying it with the coordinates of $x^2 + 2x - 2$, written as a column vector:

$$A \cdot \begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -4 \\ 1 \end{pmatrix},$$

which evaluates to the polynomial $x^2 - 4x + 1$ we found in (14). Isn't it neat that you can shift polynomials along the x-axis simply by multiplying them with a matrix – just because shifting is a linear transformation?

16 Questions to think about

≈ November

(a) At 3:47 in the first video, 3B1B says "It turns out that you only need to record where the two basis vectors ... land, and everything else will follow from that." This is essentially Proposition 8.4 in M&T. Spell out how exactly the example in the video and Proposition 8.4 relate: What are K, U, V, S, and f in the video?

(b) Also have a look at the proof of Proposition 8.4. How does it relate to 3B1B's argument of grid lines remaining parallel and evenly spaced?

(c) The start of Section 8.3 in the lecture notes contains a general definition of how to obtain a matrix for any linear transformation. Apply this definition to the example at 6:03 in the first video and check that it gives the same matrix as in the video. Hint: Use the standard basis $\{\hat{\mathbf{\imath}}, \hat{\mathbf{\jmath}}\}$.

(d) Consider the following matrices (hint: B shows up in the videos).

$$A = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Mentally (or on paper) visualize the corresponding linear transformations. Can you describe what they do with just a few words? Further, obtain the matrix AB both by calculation and by following where the basis vectors land, as in the video. Do you get the same result both ways? Can you follow $\hat{\mathbf{\imath}}$ and $\hat{\mathbf{\jmath}}$ purely visually in your head?

17 In-class activities

≈ **November**

(a) Choose a linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$, e.g. one which scales the plane by a certain factor in a certain direction, or one which rotates it around the origin by a certain angle, or both. Write down the matrix with respect to the standard basis (first column is $T(\hat{\mathbf{\imath}})$, second column is $T(\hat{\mathbf{\jmath}})$) and check that it is correct by applying it to a few vectors.

(b) Choose a matrix $A \in \mathbb{R}^{2,2}$. Use the wonderful Matrix visualization tool by Yuri Sulyma from Brown University to see what linear transformation on $\mathbb{R}^2$ that matrix describes (with respect to the standard basis). Describe the linear transformation in your own words. Try matrices with linearly independent column vectors and matrices with linearly dependent column vectors. What can you observe about the case of linearly dependent columns? (You may have to add 0.1 to one of the coefficients to see more clearly what's going on.)

18 Kernels, images and ranks ≈ November

- M&T, Sections 9
- Zach Star: Dear linear algebra students, this is what matrices (and matrix manipulation) really look like
- 3Blue1Brown: Inverse matrices, column space, and null space. Ignore all references to determinants and inverse matrices; we will talk about these next time.
- Gilbert Strang, The four fundamental subspaces. This video does a really nice job of explaining the connection of the null space, the column space, and the row space. It's an optional video if you want to learn about how these concepts relate precisely. Note that Strang focuses on $\mathbb{R}^n$ with the standard basis, and therefore doesn't distinguish between vectors and coordinates, or null spaces and kernels.
- Steve Brunton, Linear systems of equations. This is an optional video for you, which I think does a great job of explaining the relationship of null space and column space and the solvability of SLEs. It comes from the context of a course on the Singular Value Decomposition (SVD) and uses a few words that you may not know, so see if the video helps you or not. Ignore all references to SVD and the matrices U and V.

This segment of the course finally answers the question of why it is that some systems of linear equations can be solved uniquely, while others are inconsistent or underconstrained. One challenge for you is that 3B1B and M&T explain some concepts in different order, so you will have to ignore 3B1B's references to determinants and inverse matrices until we discuss them over the next two segments.

Solutions of linear equation systems. The key finding, explained e.g. at the end of Section 9 in M&T, is the following.[6]

1. An SLE $A\mathbf{x} = \mathbf{b}$ *has a solution* iff $\mathbf{b} \in \operatorname{im}(A)$. If $\mathbf{b}$ is not in the image of A, i.e. not a linear combination of its column vectors, then the SLE is inconsistent. *Homogeneous* SLEs, with $\mathbf{b} = \mathbf{0}$, always have a solution because the zero vector is in the image of any linear transformation.
2. If the SLE has any solution at all, then it has a *unique solution* iff the dimension of the kernel of A is zero. Otherwise you can take any solution $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$ and pick any nonzero vector $\mathbf{y} \in \ker(A)$, and $\mathbf{x} + \mathbf{y}$ will give you another solution: $A(\mathbf{x}+\mathbf{y}) = A\mathbf{x} + A\mathbf{y} = \mathbf{b} + \mathbf{0} = \mathbf{b}$. In the case of $\mathbb{R}^n$, all nontrivial kernels contain infinitely many vectors, and therefore you get infinitely many solutions.

This situation is summarized in Table 1.

In the case of a square matrix, say of size $n \times n$, we have $\dim(\ker(A)) + \dim(\operatorname{im}(A)) = n$, per Theorem 9.6. Therefore the dimension of the kernel is zero if and only if the

[6]Whenever I refer to the "image" or "kernel" of a matrix A, I mean the image or kernel of the corresponding linear transformation.

	$\mathbf{b} \in \mathrm{im}(A)$	$\mathbf{b} \notin \mathrm{im}(A)$
$\dim(\ker(A)) = 0$	unique	inconsistent
$\dim(\ker(A)) > 0$	underconstrained	inconsistent

Table 1: Solvability of linear equation systems. Homogeneous SLEs always have $\mathbf{b} \in \mathrm{im}(A)$.

dimension of the image (called the *rank*) is n. This means that an SLE can only have a unique solution if it has full rank (i.e., rank n). This has a number of equivalent characterizations, which follow from the considerations in Section 9.3 of M&T:

1. All column vectors are linearly independent.
2. All row vectors are linearly independent.
3. The upper echelon form has no non-zero rows.

Note the very pretty construction of Theorem 9.11, which relates the operations of the Gauss algorithm to linear combinations and linear independence.

Non-square matrices. Now consider the case of an $m \times n$-matrix A with $m \neq n$. Such a matrix corresponds to a linear transformation T from a vector space U with dimension n to a vector space V with dimension m. Then per Theorem 9.6, the rank and nullity still sum to n, the dimension of U. You can think of it like this: if you extend a basis B of the kernel of A (with $\dim(\ker(A))$ vectors in it) to a basis $B' \supseteq B$ of U (with n vectors in it), then A will send $\mathrm{span}(B)$ to zero, and the image of T will be spanned by the (linearly independent) images of $B' - B$. Therefore the dimension of the image will still be $n - \dim(\ker(A))$.

It is very cool, and not initially obvious, that the row rank and the column rank of a matrix are the same (Corollaries 9.12, 9.13). This means that even in a non-square matrix, the number of linearly independent rows is the same as the number of linearly independent rows. As a consequence:

- If you have a tall and skinny matrix, $m > n$, then the rank is at most n, and you know that when you bring the matrix into row echelon form, there will be at most n non-zero rows, and thus at least $m - n$ zero rows. This by itself does not mean that the SLE can't have a unique solution; if the matrix has rank n, meaning that *only* $m - n$ rows end up zero and not more, then the dimension of the kernel will still be zero.
- If you have a squat and wide matrix, $n > m$, then the row echelon form may not have any rows which are zero. Nonetheless, because the dimension of the image (= the rank) is at most m, the dimension of the kernel is at least $n - m \geq 1$, and thus there are either no solutions or infinitely many. In such a case, it may be useful to bring the matrix into row and column reduced form; this will create at least $n - m$ zero *columns*.

Some further notes. Corollary 9.7 uses words which you may not be familiar with. A linear transformation $T : U \to V$ is

- *injective* if for any two vectors $\mathbf{u}, \mathbf{u}' \in U$ with $\mathbf{u} \neq \mathbf{u}'$, we have $T(\mathbf{u}) \neq T(\mathbf{u}')$; that is, different vectors have different images.
- *surjective* if for every $\mathbf{v} \in V$, there is some $\mathbf{u} \in U$ such that $T(\mathbf{u}) = \mathbf{v}$; that is, every vector in V is the image of some vector in U.
- *bijective* if it is injective and surjective, i.e. a one-to-one correspondence of vectors in U and in V.

Example 9.3 and the examples after Theorem 9.6 are very important to build intuitions. Go through them, one by one, and check whether all the statements are correct. If questions come up, let's talk about them in class.

Using the terminology from our notes, Proposition 7.8 can be rephrased as "all spans are subspaces". Conversely, every subspace is a span: the subspace has a basis, and it is spanned by any of its bases.

Finally, note that the contrast between M&T and 3B1B is particularly visible here. 3B1B works with matrix multiplications and geometric intuitions, whereas M&T does hardcore abstract linear algebra and proves the theorems with no assumptions about what the vector spaces look like. I like this contrast a lot – it illustrates nicely that there are many equivalent ways to think about the same mathematical constructions, and being able to switch between intuitions is very valuable.

Howto: Determining the kernel of a linear transformation. It is easy to determine the rank and nullity of a linear transformation: For the rank, you sift either the rows or the columns of its matrix to find out how many of them are linearly independent; or you can bring the matrix into upper echelon form and count non-zero rows. The nullity is the dimension of the source space, minus the rank.

The nullity is only the *dimension* of the kernel. To find out what the kernel of A really looks like, you solve the homogeneous SLE $A\mathbf{x} = \mathbf{0}$. The solution set of this SLE is the kernel of the matrix. Kernels are always subspaces, and you can determine a basis for the kernel by looking at the solution set.

Consider, for instance, the matrix

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 3 & 7 & 2 \\ 2 & 5 & 1 \end{pmatrix}.$$

As we saw in Section 3, the upper echelon form of the SLE $A\mathbf{x} = \mathbf{0}$ is the matrix B shown in (3). We can rewrite the solution set (6) as follows:

$$S = \{x_3 \cdot \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix} \mid x_3 \in \mathbb{R}\} = \mathsf{span}(\left\{ \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix} \right\}).$$

S is the set of all vectors from $\mathbb{R}^3$ that are mapped to zero by A, so S is the kernel of A. Observe that we have expressed it as a linear combination of vectors, with the scaling factor of each vector being one of the free variables of the SLE. The vectors thus span the kernel. In fact, they are linearly independent, so $\{[-3, 1, 1]\}$ is a basis of the kernel.

Note that the size of this basis is the same as the dimension of the kernel, which you can easily compute from the rank. This means, among other things, that the number of free variables in the upper echelon form of an SLE is the same as the nullity of the matrix.

19 Questions to think about

≈ November

(a) Prove Theorem 9.14 in M&T.

20 In-class activities

≈ November

(a) Solve a few random SLEs $A\mathbf{x} = \mathbf{b}$ (or look up their solutions online). Determine the rank and nullity of A. Double-check that the SLEs behave as described in Table 1.

(b) Pick a few 2×2 and 3×3 matrices, perhaps the ones you used in (a), and determine their kernels, both in set and in span-of-basis notation. Be sure to look at matrices with nullity 1 and 2; construct matrices with sufficiently low rank by hand if necessary.

21 Inverse matrices and determinants ≈ November

- ▶ 3Blue1Brown: The determinant
- ▶ 3Blue1Brown: Inverse matrices, column space, and null space
- ▶ M&T, Sections 10–11.3
- ▶ Optionally, read Section 11.4–11.6 and watch 3Blue1Brown: Cramer's Rule.
- ▶ Practice: Random matrix inversion problems
- ▶ Practice: Random determinant problems (2x2, 3x3)

In this segment, we introduce the concepts of *inverting* a matrix and of its *determinant*. The two concepts apply only to square matrices and are linked by Theorem 11.11 in M&T: A square matrix can be inverted if and only if its determinant is nonzero. This is because a linear transformation has a zero determinant iff its image has lower dimension than its source space; such linear transformations can't be inverted, because you can't surjectively "unsquish" the lower-dimensional image back into the higher dimension with a linear transformation.

Notice that inverting matrices is in principle the same as inverting rational numbers: A^{-1} is the matrix such that $A^{-1}A = I$, just like $3^{-1} = 1/3$ is the number such that $3^{-1} \cdot 3 = 1$. However, unlike with rational numbers, some matrices don't have an inverse.

In M&T, a formula for the determinant of a matrix is presented. In this course, we will not go into detail about this formula and focus more on the geometric interpretation and general properties of determinants.

Howto: Computing inverses and determinants. M&T explain how to invert a matrix in Section 10.2.

To compute the determinant of a matrix, you can bring the matrix into upper triangular form using row operations; then the determinant is the product of the values on the diagonal (Example 11.7). **Note:** Some row operations change the determinant! If you flip rows, you flip the sign of the determinant; if you scale a row, you scale the determinant. See Theorem 11.3 in M&T for details.

Alternatively, you can use cofactors (Example 11.20). The method with row operations may be preferrable, because row operations are probably second nature to you at this point in the course, and you are less likely to make mistakes.

Formulas for inverses and determinants. In general, one can derive formulas for the determinant and the inverse of a matrix. These become very complicated for large n but might be useful in the case of 2×2- and 3×3-matrices:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \qquad \det(A) = ad - bc, \qquad A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix},$$

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}, \qquad \det(A) = aei + bfg + dhc - ceg - bdi - fha,$$

$$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} ei - fh & ch - bi & bf - ce \\ fg - di & ai - cg & cd - af \\ dh - eg & bg - ah & ae - bd \end{pmatrix}.$$

Optional: Cramer's Rule. Sections 11.5 and 11.6 explain a method for inverting matrices and for solving SLEs using determinants. These are nice connections, and Cramer's Rule has a geometric interpretation that is explained in the 3B1B video. Feel free to explore these connections if you're interested; we won't talk about them in this class.

22 Questions to think about

≈ November

(a) You can solve any SLE $A\mathbf{x} = \mathbf{b}$ by computing A^{-1} and then $\mathbf{x} = A^{-1}\mathbf{b}$. Why is this an inefficient way of solving the SLE?

(b) Explain, in terms of rank and nullity, the connection between the invertibility of A and the number of solutions of the SLE $A\mathbf{x} = \mathbf{b}$.

(c) Explain in one sentence, with an appeal to the geometric intuition from the 3B1B video, why $\det(AB) = \det(A)\det(B)$. Consider, maybe in a second sentence, what happens if one or both determinants are negative.

(d) Consider the following matrix:

$$\begin{pmatrix} 2 & -2 \\ 1 & -1 \end{pmatrix}$$

Looking at only the determinant of the matrix, what can you say about its rank? The dimension of its kernel? Can the matrix be inverted?

(e) We have now seen how matrices can be added, multiplied, and inverted. Check which field axioms from M&T, Section 4.1 are satisfied by matrices and which ones are not.

23 In-class activities

≈ November

(a) Using the online tool, generate a few random 3x3 matrices and invert them.

(b) Using the online tool, generate a few random 2x2 matrices and compute their determinants. Also generate a couple of 3x3 matrices and compute their determinants.

24 Change of basis

≈ **November**

- 3Blue1Brown: Change of basis
- M&T, Section 12

Coordinates of vectors. Back in Section 15, we talked in a lot of detail about the distinction between a vector and its coordinates with respect to a basis. At the time, you probably thought this was overkill – after all, in $\mathbb{R}^n$ with the standard basis, the components of the vector are always exactly the same as its coordinates.

In this section, it is now the whole point that the matrix of a linear transformation changes when you change the basis. This is really hard to wrap your head around when you think of a matrix as transforming vectors. The correct way to think about a matrix is that it *transforms coordinates*, not vectors. Thus, if you change the basis, the coordinates of all vectors change; and therefore the matrix must change along with them so they can transform the new coordinates accordingly.

You may want to go gack to Section 15 and reread the paragraph titled "Representing linear transformations as matrices". If anything about this is unclear to you, you will struggle with change of basis. So let's talk about it in class or over Classroom.

Change of basis. The key technical point of this section is that we can describe neatly and succinctly how the matrix of a linear transformation changes when we change one or both bases. If you look at Fig. 1, you have a fixed linear transformation $T : U \to V$. Expressed with respect to the bases $E = \{\mathbf{e}_i\}_i$ for U and $F = \{\mathbf{f}_i\}_i$ for V, we can represent T with some matrix A. If we choose different bases $E' = \{\mathbf{e}'_i\}_i$ for U and $F' = \{\mathbf{f}'_i\}_i$ for V, we obtain a different matrix B.

Theorem 12.4 says that if P is the change-of-basis matrix from E to E' and Q is the change-of-basis matrix from F to F', we have $B = QAP^{-1}$. P maps coordinates in U with respect to E to coordinates in U with respect to E' (of the same vector, because P represents the identity transformation); thus P^{-1} translates E'-coordinates to E-coordinates. These E-coordinates can be transformed to F-coordinates via multiplication with A, because A was based on E and F. Finally, we decode these F-coordinates into F'-coordinates using Q. Thus the overall matrix encodes T with respect to inputs in terms of E'-coordinates and outputs in terms of F'-coordinates.

Which way does the change of basis go? Just like 3B1B, the definition always felt backwards to me. Personally, I have to memorize that the columns are the coordinates of the "old" basis in terms of the "new" basis. This makes sense because the first "old" basis vector has the coordinates $\langle 1, 0, \ldots, 0\rangle$ with respect to the "old" basis, and thus it is mapped to the first column of the matrix. When we evaluate this column to a vector with respect to the "new" basis, we need to get the first "old" basis vector back.

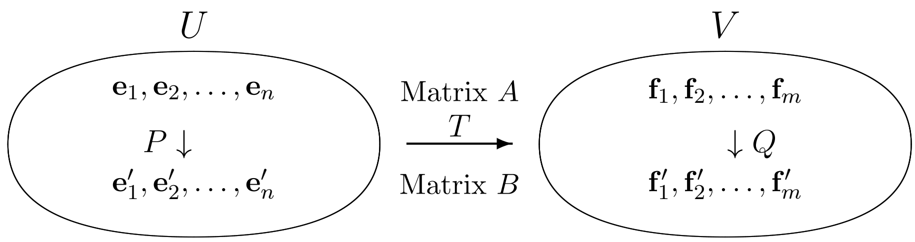


Figure 1: Change of basis; picture from M&T, page 51.

Let's look at an example. Let $E = \{\mathbf{e}_1, \mathbf{e}_2\}$ be the basis of $\mathbb{R}^2$ consisting of the following vectors:

$$\mathbf{e}_1 = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \qquad \mathbf{e}_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

Let $E' = \{\hat{\mathbf{\imath}}, \hat{\mathbf{\jmath}}\}$ be the standard basis. Then the coordinates of $\mathbf{e}_1$ with respect to E' are $\langle 0, -1\rangle$, and the coordinates of $\mathbf{e}_2$ with respect to E' are $\langle -1, 0\rangle$. We obtain the following change-of-basis matrix P from E to E':

$$P = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}.$$

Now let $\mathbf{v}$ be a vector whose E-coordinates are $\langle v_1, v_2\rangle$. P will map these coordinates as follows:

$$P \cdot \langle v_1, v_2\rangle = \langle -v_2, -v_1\rangle.$$

So for instance, $\mathbf{e}_1 \in \mathbb{R}^2$ has the E-coordinates $\langle 1, 0\rangle$, and P maps these into $\langle 0, -1\rangle$. These are just the E'-coordinates of $\mathbf{e}_1$. More generally, we can remember the direction of the change-of-basis matrix as follows:

The change-of-basis matrix from E to E' maps
E-coordinates to E'-coordinates.

If you like to think in terms of "decoding" a foreign "language", you can say that P decodes the E-coordinates of a vector into its E'-coordinates. Obviously, P^{-1} decodes E'-coordinates back to E-coordinates.

3B1B proposes a perspective at 7:00 that he finds helpful: that you can "misinterpret" a coordinate tuple as a vector with respect to the "old" basis, and then the linear transformation encoded by the matrix will map it to the correct interpretation in terms of the "new" basis. For me, this interpretation clashes too much with the clean distinction between a vector and its coordinates. For you, 3B1B's intuition may or may not be helpful; but if you take it on board, make sure you do it with caution, and respect the vector/coordinate distinction.

Summary. Change-of-basis matrices are an idea that feels complicated, but it really isn't, as long as you think about vector spaces correctly. You need to free yourself from the idea that vectors "are" tuples of numbers. The vector exists as an object in the vector space, independently of the basis that you're looking at; "space has no grid". Then when you choose a basis, this defines a coordinate system, and you can express vectors as tuples of numbers, namely coordinates. If you use $\mathbb{R}^n$ with its standard basis $\hat{\mathbf{i}}, \hat{\mathbf{j}}, \ldots$, the vectors literally are tuples of numbers, and their coordinate tuples are the same tuples of numbers, so it is easy to confuse the two views. But there is nothing special about the standard basis, and any other basis defines its own coordinate system, which is equally valid. In vector spaces that are not $\mathbb{R}^n$, the vectors aren't tuples of numbers in the first place, and yet they can still be represented using their coordinates with respect to a basis.

Howto: Computing change-of-basis matrices. To compute the change-of-basis vector from a basis E to a basis E' as above, you need to express each vector in E as a linear combination of E'. You can do this by solving linear equation systems, as explained in Section 8.

Change-of-basis matrices from some basis E to the standard basis are particularly simple: their column vectors are simply the coordinates of the basis vectors in E (see e.g. 3B1B at 5:20). You can get a change-of-basis matrix from the standard basis to E by matrix inversion (but solving the SLE directly may be cheaper).

25 Questions to think about $\approx$ **November**

(a) Remember 3B1B's question from the very first class: What is a vector? Discuss the extent to which vectors are tuples of numbers.

(b) Explain why all change-of-basis matrices are invertible.

26 In-class activities $\approx$ **November**

(a) Pick a linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$ and represent it as a matrix with respect to the standard basis $E = F = \{\hat{\mathbf{i}}, \hat{\mathbf{j}}\}$ on both sides. Then pick bases E', F' for the two vector spaces. It is a good idea to use either $E' = E$ or $F' = F$, otherwise you will have to solve a lot of SLEs. Tip: You can pick a nice basis by using the columns of a random 2x2 matrix with nonzero determinant, which you can generate with the online tool.

Compute the matrix of T with respect to E' and F', in the following ways:

(i) directly, i.e. by expressing the values of the vectors in E' under T with respect to F';

(ii) by computing the appropriate change-of-basis matrices between E and E' and between F and F' and applying Theorem 12.4.

Compare your results.

27 Eigenvalues and eigenvectors

≈ **November**

- 3Blue1Brown: Eigenvectors and eigenvalues
- Zach Star: The applications of eigenvectors and eigenvalues
- M&T, Section 13 (without Section 13.1)
- Practice: Random eigenvector problems

Eigenvalues and eigenvectors are tools that are very useful in practice (see e.g. Zach Star's video) and have very nice geometric intuitions (see e.g. 3B1B's video).

The key result in this segment is Corollary 13.14 in M&T: If a linear transformation T has n distinct eigenvalues, then it can be represented by a diagonal matrix D; the entries on the diagonal are the eigenvalues. This matrix representation is only valid with respect to the eigenbasis, but you can always take the change-of-basis matrix P from the eigenbasis to the standard basis, and then $A = PDP^{-1}$ will be the matrix of T with respect to the standard basis.

Diagonal matrices are really pleasant to work with. Matrix multiplication becomes quick and easy because most coefficients are 0, and computing the power D^n of a diagonal matrix D simply amounts to computing the n-th power of the values on the diagonal. This is really nice when you want to transform a vector through multiple applications of T. As Zach Star illustrates, this sort of thing comes up in all sorts of problems involving time, where the linear transformation describes how some state of the world changes from one timestep to the next, and you want to project the state into the distant future.

It is a regrettable fact that not all matrices can be diagonalized via their eigenbases, e.g. because some eigenvalues occur multiple times or because some roots of the characteristic polynomial are complex numbers. For such cases one can look at the Singular Value Decomposition (SVD) of a matrix, a generalization of the eigendecomposition.

Howto: Computing eigenvalues and eigenvectors. To compute the eigen-things of a matrix A, you first compute the eigenvalues, and then you compute an eigenvector for each distinct eigenvalue. M&T discuss an example in Example 13.7.

According to Theorem 13.5 of M&T, the eigenvalues $\lambda_1, \ldots, \lambda_k$ are the solutions of the *characteristic equation* of A,

$$\det(A - \lambda I) = 0.$$

When you spell this out for a specific A, you will obtain a polynomial in λ (the characteristic polynomial of A), and you need to find the values for λ that make it zero. The degree of the polynomial depends on the size of the matrix. In this course, we will only compute eigenvalues for matrices in $\mathbb{R}^{2,2}$, so computing the determinant is easy, and you will always obtain quadratic equations. You probably learned how to solve quadratic equations in high school; feel free to look up the formula on Wikipedia if you need to.

Notice that a quadratic equation has at most two solutions. It is possible that there is only one solution (e.g. $x^2 - 2x + 1 = 0$) or no solutions at all (e.g. $x^2 + 1 = 0$).[7] In these cases, there will be fewer than two eigenvalues.

The eigenvectors are those vectors $\mathbf{v}$ for which you have $A\mathbf{v} = \lambda_i \mathbf{v}$ for one of the eigenvalues λ_i. You can therefore compute them by solving the SLE

$$(A - \lambda_i I)\mathbf{v} = \mathbf{0}$$

for $\mathbf{v}$. You get a potentially different eigenvector for each eigenvalue λ_i.

Howto: Diagonalization. The diagonalization result in Corollary 13.14 is not only beautiful and useful, but the diagonalization is actually quite easy to compute.

Let $A \in \mathbb{R}^{n,n}$ be the matrix we want to diagonalize. Say that it is the matrix of the linear transformation $T : \mathbb{R}^n \to \mathbb{R}^n$ with respect to the standard basis. First, you compute the eigenvalues $\lambda_1, \ldots, \lambda_n$ with their corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$. If there are n distinct eigenvalues, you know from Theorem 13.13 that the $\mathbf{v}_i$ are linearly independent; this is the condition of Corollary 13.14. If you do not get n distinct eigenvalues, it may still be possible to diagonalize the matrix using the construction below (see the discussion in M&T after Example 13.15); you will have to check by hand if there are n linearly independent eigenvectors. A set of n linearly independent eigenvectors is called an *eigenbasis* of T.

The matrix of T given the eigenbasis $E = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ is a diagonal matrix D with the eigenvalues on the main diagonal, by definition of an eigenvector. This is how you get a diagonal matrix for T / diagonalize A.

D is connected to A in the following way. Let P be the change-of-basis matrix from the eigenbasis E to the standard basis; it will have $\mathbf{v}_i$ in the i-th column. Now all you have to do is invert P, and you will have

$$A = PDP^{-1}.$$

That is, you first take coordinates with respect to the standard basis and convert them to E-coordinates, using P^{-1}; then you perform the linear transformation with respect to E; and then you decode the E-coordinates back to the standard basis using P.

Of course, you can also think about this the other way round: If you have A and P, you can diagonalize A by the opposite change of basis,

$$D = (P^{-1}P)D(P^{-1}P) = P^{-1}(PDP^{-1})P = P^{-1}AP.$$

(Isn't the law of associativity nice?) But given that you need to know the eigenvalues of A to compute P anyway, you might just as well write down D as the matrix with the eigenvalues on the diagonal in the first place.

[7] The latter equation *does* have a solution if the solution can be a complex number, namely $x = \pm i$. In fact, the Fundamental Theorem of Algebra says that *every* nontrivial polynomial has at least one complex solution. This is why discussions of eigenvalues drift into complex numbers quite easily. Here we focus only on solutions that are real numbers.

28 Questions to think about ≈ **November**

(a) Solve the "puzzle" at the end of the 3B1B video (at 16:30). Given the matrix

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix},$$

first figure out by hand what sequence the top-right coefficient of $A^1, A^2, A^3, \ldots$ represents; it is a famous sequence of numbers. Then use the fact that eigenvectors give you an easy way to compute A^n to obtain a closed-form expression for the elements of that sequence. The eigenvectors of A are $\mathbf{v}_1 = [2, 1+\sqrt{5}]$ and $\mathbf{v}_2 = [2, 1-\sqrt{5}]$.

Here are some hints:

- As you know from high school, $(a+b)(a-b) = a^2 - b^2$. Also remember the binomial theorem.
- The eigenvalues for $\mathbf{v}_1$ and $\mathbf{v}_2$ are $1+\sqrt{5}$ and $1-\sqrt{5}$, respectively. (Don't take my word for it, compute them yourself!)
- Write down the change-of-basis matrix P from the eigenbasis to the standard basis. Then compute P^{-1} in your preferred way; you will have $A = PDP^{-1}$. Hint: P^{-1} will be $\frac{1}{4\sqrt{5}}$ times a matrix with reasonable coefficients.
- Now you can compute $A^n = PD^nP^{-1}$, with D a diagonal matrix containing the eigenvalues. It is relatively straightforward to compute D^nP^{-1}; then when you multiply that with P on the left, feel free to compute only the top right coefficient (because this is the number we care about).

29 In-class activities ≈ **November**

(a) Play with Yuri Sulyma's matrix visualizer to get a geometric sense of eigenvalues and eigenvectors. Plug in different linear transformations, observe in what directions the picture gets stretched and squeezed, and notice how this relates to the eigenvectors drawn in blue and red. Notice that linear transformations in $\mathbb{R}^2$ can have zero, one, or two eigenvectors, and what happens when you have eigenvalues other than 1.

(b) Pick a few linear transformations in $\mathbb{R}^2$ – perhaps ones with nice geometric interpretations – and compute their eigenvectors and eigenvalues.

(c) Where eigenbases are available, work out the change-of-basis matrices and write the matrix of the linear transformation as a product PDP^{-1}, where D is a diagonal matrix. Check that calculating the matrix product actually gives you the original matrix back.

30 Dot products

- 3Blue1Brown: Dot products and duality
- 3Blue1Brown: Cross products
- 3Blue1Brown: Cross products in the light of linear transformations
- M&T, Section 13.1

In this segment, we look at how geometric notions like lengths and angles can be defined on $\mathbb{R}^n$. This is cool, because it is clear what lengths and angles mean in $\mathbb{R}^2$; but now we can compute lengths and angles in very high-dimensional vector spaces that we could never draw.

M&T do not talk about dot products in detail. Feel free to follow some of the links below to Wikipedia, where these concepts are explained quite well.

Dot products. Let

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \text{ and } \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

be two vectors in $\mathbb{R}^n$. Then their *dot product* (or *scalar product*, or *inner product*) is defined as

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n. \tag{15}$$

An equivalent way to think about this is to say that $\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v}$, where $\mathbf{u}^T$ is the row vector version of $\mathbf{u}$, and $\mathbf{u}^T \mathbf{v}$ is simply matrix multiplication: if you think of $\mathbf{u}^T$ as a $1 \times n$-matrix (consisting of a single row) and $\mathbf{v}$ as an $n \times 1$-matrix (consisting of a single column), then $\mathbf{u}^T \mathbf{v}$ is a 1×1-matrix containing a single scalar.

Observe that the dot product is a way of multiplying two vectors, which seems to contradict the point we made very forcefully in Section 2, that there is no vector-vector multiplication in a vector space. However, note that the dot product of two vectors is not a vector: it is an element of the underlying field, e.g. a real number.

30.1 Geometry: lengths and angles

Norms. Dot products can be used to define lengths and angles in $\mathbb{R}$-vector spaces. The *norm* of a vector $\mathbf{u}$,

$$\|\mathbf{u}\| = \sqrt{\mathbf{u} \cdot \mathbf{u}}, \tag{16}$$

represents the length of the vector $\mathbf{u}$. This is easy to see in $\mathbb{R}^2$, because if you spell out the definition of the dot product (do it!), you will find that this is just the Pythagorean theorem.

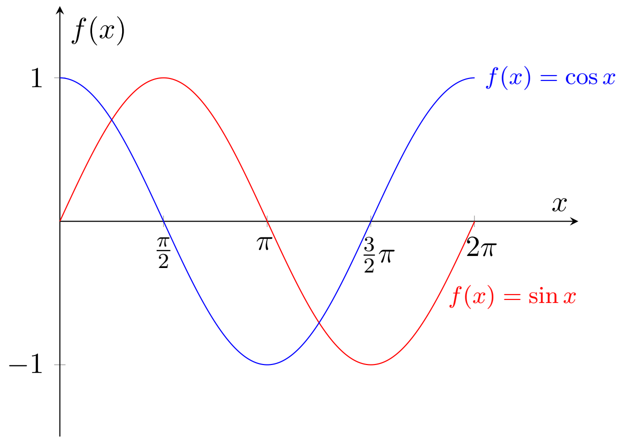


Figure 2: Sine and cosine.

The *Euclidean distance* between $\mathbf{u}$ and $\mathbf{v}$ is defined as $\|\mathbf{u} - \mathbf{v}\|$. The Euclidean distance is typically just referred to as "distance" and matches our intuition on distances in space, when we interpret the vectors as points.

We sometimes say that a vector $\mathbf{v}$ has *unit length* if $\|\mathbf{v}\| = 1$. We can *normalize* any vector $\mathbf{v} \neq \mathbf{0}$ into a vector $\mathbf{v}'$ of unit length by letting

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

Angles. When we interpret vectors in $\mathbb{R}^n$ as arrows from the origin to the given coordinates, then we can measure the *angle* between these arrows.

The angle α between $\mathbf{u}$ and $\mathbf{v}$ satisfies the following formula:

$$\cos\alpha = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \, \|\mathbf{v}\|}$$

The angle in this context is typically expressed as radians rather than degrees. Have a look at this video if you are unfamiliar with radians.

Given the fraction on the right, you can recover the angle (in radians, as a number between 0 and π) by using the arccosine function, which is the inverse of the cosine function:

$$\alpha = \arccos\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \, \|\mathbf{v}\|}.$$

You can compute the arccosine e.g. with the Python function math.acos.

The value $\cos\alpha$ is sometimes also referred to as the *cosine similarity* of $\mathbf{u}$ and $\mathbf{v}$. If the angle between $\mathbf{u}$ and $\mathbf{v}$ is below $\frac{\pi}{2}$ or 90°, e.g. if all coordinates in the vectors are non-negative, then the cosine is a number between 0 and 1 (see Fig. 2). A cosine of 1 corresponds to an angle of 0° (radians 0), i.e. the vectors point in the same direction; a cosine of 0 corresponds to an angle of 90° (radians $\frac{\pi}{2}$), i.e. the vectors are *orthogonal.* This is why it makes sense to think of $\cos \alpha$ as a similarity measure. It is used very frequently like that in distributional semantics and in the analysis of neural networks as a similarity measure between vectors.

Inner products. If you want to be very precise, the dot product defined in (15) is only one example of an *inner product*. More generally, an inner product over some $\mathbb{R}$-vector space V is a function $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ that satisfies a number of axioms. In particular, an inner product needs to be linear in both arguments; symmetric (i.e. $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$); and non-negative. You can go through the axioms on Wikipedia and convince yourself that the dot product as defined above is actually an inner product.

A vector space with an inner product is called an *inner product space*. Thus, the vector space $\mathbb{R}^n$ with the dot product defined in (15) is one example of an inner product space; it is also called the Euclidean space of dimension n. Each inner product defines a norm $\|\mathbf{u}\| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$. If you want to be precise, the norm defined in (16) is called the *Euclidean norm*; in machine learning papers, you will also find the name *L2-norm*.

30.2 Orthogonality

Orthonormal bases. Observe that Definition 13.17 in M&T defines an *orthonormal* basis of an inner product space as a basis $E = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ in which $\langle \mathbf{v}_i, \mathbf{v}_i \rangle = 1$ for all i and $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$ for all $i \neq j$. Vectors with the second property are called *orthogonal*; this is consistent with the geometric concept of orthogonality you know from high school because it means that the angle between $\mathbf{v}_i$ and $\mathbf{v}_j$ is $\frac{\pi}{2}$ or 90° (see Fig. 2). The first condition implies that every vector in the basis has length 1 (has been "normalized"). This explains the name "orthonormal".

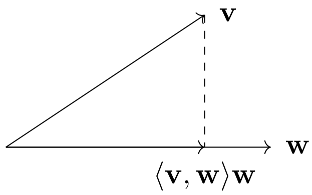
It can be proved that every finite-dimensional inner product space has an orthonormal basis (using the Gram-Schmidt process). The standard basis $\hat{\mathbf{\imath}}, \hat{\mathbf{\jmath}}, \ldots$ of $\mathbb{R}^n$ is the most prominent example of an orthonormal basis.

Orthogonality and linear independence. In $\mathbb{R}^n$, we have the very clear intuition that if a set of vectors are all pairwise orthogonal, then that set is linearly independent. But orthogonality and linear independence are actually two very different concepts: one is defined in terms of the inner product, the other in terms of linear combinations. It is therefore a really cool little result that orthogonality implies linear independence, both in $\mathbb{R}^n$ and in any other inner product space.

Orthogonal projections. 3Blue1Brown gives a nice explanation of how the inner product $\langle \mathbf{v}, \mathbf{w} \rangle$ can be interpreted as the length of the orthogonal projection of $\mathbf{v}$ onto $\mathbf{w}$ times the length of $\mathbf{w}$. Hence, if $\mathbf{w}$ has length 1 then $\langle \mathbf{v}, \mathbf{w} \rangle$ computes only the length of the orthogonal projection of $\mathbf{v}$ onto $\mathbf{w}$. Further, if we scale $\mathbf{w}$ by this length then we obtain the vector

$$\langle \mathbf{v}, \mathbf{w} \rangle \mathbf{w},$$

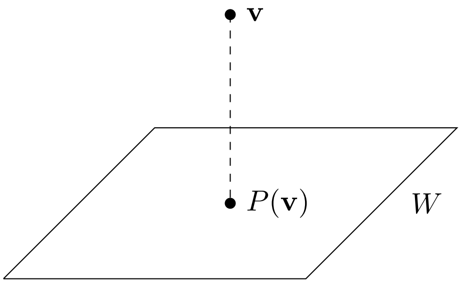
which corresponds to the vector $\mathbf{v}$ projected down onto the line spanned by $\mathbf{w}$.



More generally, if $W \subseteq V$ is a subspace of an inner product space V and if $\mathbf{w}_1, \ldots, \mathbf{w}_n$ are an orthonormal basis of W then

$$P(\mathbf{v}) = \sum_{i=1}^{n} \langle \mathbf{v}, \mathbf{w}_i \rangle \mathbf{w}_i$$

defines the orthogonal projection onto the subspace W.



For a given vector $\mathbf{v}$ the projection $P(\mathbf{v})$ has the nice property that it is the vector $\mathbf{w} \in W$ which minimises the distance $\|\mathbf{v} - \mathbf{w}\|$. Solving minimization problems using this approach has many applications and gives for example linear regression and Fourier series a nice geometric interpretation.

30.3 The dual space

The dot product is one example of a linear form, i.e. of a linear transformation $T : \mathbb{R}^n \to \mathbb{R}$, where we look at both $\mathbb{R}^n$ and $\mathbb{R}$ $(= \mathbb{R}^1)$ as $\mathbb{R}$-vector spaces. There is a very cool *duality* between the set of linear forms and the set of vectors in $\mathbb{R}^n$ which is mediated by the dot product. This is discussed in 3B1B's videos and in Section 8.2 of M&T.

First, take some vector $\mathbf{v} \in \mathbb{R}^n$. We can define a function $T_{\mathbf{v}} : \mathbb{R}^n \to \mathbb{R}$ by letting $T_{\mathbf{v}}(\mathbf{u}) = \mathbf{v} \cdot \mathbf{u}$. As you can easily check, $T_{\mathbf{v}}$ is a linear form. Thus, you can encode each vector into a function – a function that maps vectors into scalars by evaluating the dot product.

Conversely, and this is the cool part, if you take any linear form $T : \mathbb{R}^n \to \mathbb{R}$, there is a unique vector $\mathbf{v}_T \in \mathbb{R}^n$ such that $T(\mathbf{u}) = \mathbf{v}_T \cdot \mathbf{u}$ for all $\mathbf{u} \in \mathbb{R}^n$. Thus, any linear transformation into $\mathbb{R}$ can be encoded as a single vector. This seems a bit magical, until you remember that the transpose of $\mathbf{v}_T$ is simply the $1 \times n$-matrix of the linear transformation T with respect to the standard basis, and the dot product $\mathbf{v}_T \cdot \mathbf{u}$ corresponds to matrix-vector multiplication with a matrix that happens to contain just one row.

This equivalence between the elements of an inner product space and its *dual space* (the space of linear forms) is extremely useful in many fields of science, such as theoretical physics. It also enables very pretty constructions, such as the one for the cross product in 3B1B's videos. Cross products are important in physics, but they only really make sense in $\mathbb{R}^3$, and so we don't use them much in the high-dimensional vector spaces that are relevant in machine learning and NLP.

31 Questions to think about ≈ December

(a) What other orthonormal bases are there of $\mathbb{R}^2$, other than the standard basis $\{\hat{\mathbf{\imath}}, \hat{\mathbf{\jmath}}\}$?

(b) How big can a set of vectors in $\mathbb{R}^n$ be such that they are all pairwise orthogonal? Think about $\mathbb{R}^2$ and $\mathbb{R}^3$ first. Then generalize your findings to higher n and think about why your theorem is true.

(c) What is the dual matrix for the column vector

$$\mathbf{v} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \in \mathbb{R}^2?$$

That is, the matrix with respect to the standard basis corresponding to the linear transformation $T : \mathbb{R}^2 \to \mathbb{R}$ such that $T(\mathbf{u}) = \mathbf{v} \cdot \mathbf{u}$ for all $\mathbf{u}$?

32 In-class activities ≈ December

(a) Pick a few vectors in $\mathbb{R}^2$ and $\mathbb{R}^3$ and compute their dot products and lengths. Using a ruler, double-check for some vectors in $\mathbb{R}^2$ that their measured length is the same as the length you compute using linear algebra.

(b) Normalize the vectors in (a) that didn't already have unit length. Draw them and inspect them geometrically.

(c) Pick a few vectors in $\mathbb{R}^2$ and compute the angles between them. Draw the vectors and measure the angles, e.g. with a Geodreieck. Compare your results. Be sure to also look at angles greater than 90°.

33 (Linear algebra with Numpy) ≈ December

- Numpy documentation
- TutorialsPoint: Numpy – Linear Algebra (dot product, matrix multiplication, determinants, inverting matrices, SLEs)
- Towards Data Science: Introduction to linear algebra with Numpy (basics)
- GeeksforGeeks: Numpy / Linear Algebra (ranks, determinants, eigenvalues, multiplication, SLEs)
- Stanford CS231A: Python introduction and linear algebra review

To conclude Part 1 of this course, let's have a look at numpy, a Python library for manipulating vectors and matrices. All major math-related libraries for Python, including the Pytorch and Tensorflow neural networks libraries, have interfaces to Numpy and are modeled after it, and it is definitely worth learning. Among many other functions, Numpy has functions for doing everything we did by hand so far, by computer.

There isn't one convincing and complete Numpy tutorial on the Internet that I could point to, so I have linked a few that will each give you part of the story. Note that the geeksforgeeks tutorial is overall quite nice, but you need to ignore their somewhat self-important decision to randomly rename some imports to `geek`. The convention is to always `import numpy as np`.

Here are some things you can do with Numpy:

- Create arrays to represent matrices and vectors. Note that you can either explicitly specify the values, or you can fill the array with zeroes or ones or random numbers or build some special matrix (e.g. the identity matrix).
- Address cells and take slices of matrices and vectors, e.g. the column and row vectors of a matrix.
- Solve linear equation systems.
- Multiply matrices with each other and with vectors using the matmul function
- Compute the rank of a matrix. Use this to check vectors for linear independence.
- Invert matrices and compute their determinants.
- Compute the eigenvalues and eigenvectors.
- Compute the length of a vector. By computing dot products, you can easily write your own function to compute angles.

34 In-class activities ≈ December

(a) Play around with Numpy's methods for creating, modifying, and indexing arrays.

(b) Redo a few exercises from the course using Numpy instead of by hand.