Warwick Mathematics Exchange

MA106

Linear Algebra

2022, July 20th

Desync, aka The Big Ree

Contents

Introduction

Linear algebra is the study of systems of linear equations, vector spaces and linear maps. Linear algebra is essential to almost every area of mathematics, due to how abstractly vector spaces are defined. If your problem/model/whatever has some notion of scaling things, and adding two things together to get a third, linear algebra theorems probably apply to it.

In this course, we focus mainly on matrices and vector spaces. This module is very computational in nature, with many questions simply asking you to perform simple calculations. This computational aspect is shared with MA133 *Differential Equations*. However, the more complex questions that will be asked of you require a much stronger understanding of vector spaces compared to what you need to know about differential equations for MA133.

This document is intended to closely cover all the topics within the Linear Algebra module in a much more intuitive way than was taught in my year. Matrix algebra skills from A-Levels not assumed, and we will rederive the definition of matrix-vector and matrix-matrix multiplication from geometric principles. Knowledge of eigenvalues and eigenvectors (from Edexcel FP2 o.e.) is not assumed, but prior experience is certainly helpful.

This document is not designed to be a replacement for lecture notes, though this revision guide is far more comprehensive than most of my others, and is far more usable as a replacement, if you do decide to use it as such. Much of the content here is covered in a different order than is taught in the course (for example, the Gaussian elimination algorithm is largely skipped over, since the specifics doesn't really matter, as long as you know how to row reduce a matrix), but is written in a way to maximise intuition and understanding.

I certainly recommend you still look at the notes from your year, as this document will focus more on intuition and may therefore pass over some technical details rather quickly.

If you do wish to learn most of this module outside of lectures and reading lecture notes, I recommend watching this series by Grant Sanderson. This playlist covers almost everything you need in terms of background knowledge. Much of this document will simply rehash that series (in a slightly different order), with extra notes about computation added in that Sanderson tends to omit in favour of visual intuition.

Many of the techniques developed here are used extensively in further (and many core!) modules - notably, MA136, MA249, MA251, and MA377, so take care not to just forget everything you've learned once the exam has passed by.

As some of you will just want a quick list of results and methods, I have also included a condensed summary at the end, as with many of my other guides.

Disclaimer: This document was made by a first year student who stopped going to lectures for this module after week 2. I make *absolutely no guarantee* that this document is complete nor without error. In particular, any content covered exclusively in lectures (if any) will not be recorded here. Additionally, this document was written at the end of the 2022 academic year, so any changes in the course since then may not be accurately reflected.

Notes on formatting

New terminology will be introduced in *italics* when used for the first time. Named theorems will also be introduced in *italics*. Important points will be **bold**. Common mistakes will be underlined. The latter two classifications are under my interpretation. YMMV.

Content not taught in the course will be outlined in the margins like this. Anything outlined like this is not examinable, but has been included as it may be helpful to know alternative methods to solve problems.

The table of contents above, and any inline references are all hyperlinked for your convenience.

Scalars are written in lowercase italics, c, or using greek letters.

Vectors are written in lowercase bold, $\mathbf{v}$, or rarely overlined, $\overleftrightarrow{v}$, where more contrast or clarity is required.

Matrices are written in uppercase bold, $\mathbf{A}$.

Note: transformations represented by matrices may be written in just italics, as functions often are, i.e., $s(\mathbf{v}) = \mathbf{A}\mathbf{v}$.

I do not recommend using a greyscale version of this document. I have tried being colourblind friendly wherever possible (adding dotted/dashed contrast elements, labels, avoiding green/red together, etc.), but I myself am not colourblind. If you have trouble deciphering some of my diagrams, please get in touch and I will be happy to push an update.

History

First Edition: 2022-06-12[*]
Current Edition: 2022-07-20

Note to future maintainers: the techniques I have used to draw the transformed grids is extremely temperamental, and I have had to manually adjust the scaling and positioning of every single grid. I absolutely do not recommend touching any of that code unless you really want to redo every single matrix transformation. Even resizing the grids in the axis environments will cause issues with alignment, so wrap everything inside a resize box instead if needed (though this will screw up the size of any text present). Please take care with reformatting.

Authors

This document was written by R.J. Kit L., a maths student. I am not otherwise affiliated with the university, and cannot help you with related matters.

Please send me a PM on Discord @Desync#6290, a message in the WMX server, or an email to Warwick.Mathematics.Exchange@gmail.com for any corrections. (If this document somehow manages to persist for more than a few years, these contact details might be out of date, depending on the maintainers. Please check the most recently updated version you can find.)

If you found this guide helpful and want to support me, you can buy me a coffee!

(Direct link for if hyperlinks are not supported on your device/reader: ko-fi.com/desync.)

[*]Storing dates in big-endian format is clearly the superior option, as sorting dates lexicographically will also sort dates chronologically, which is a property that little and middle-endian date formats do not share. See ISO-8601 for more details. This footnote was made by the computer science gang.

1 Vectors

1.1 Mathematical Interpretation

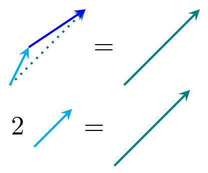
In physics, vectors are often treated as *arrows* pointing in space - some kind of quantity which has a *magnitude* and a *direction.* As long as the length and direction of a vector are the same, it's the same vector, no matter where it is. For example, you might model the velocity of an object as a vector, and consider the velocity as staying the same if the length and direction remain constant.

Sets of vectors that all lie within a plane are two-dimensional, and those in the space we live in are three-dimensional. Picturing an arbitrary n-dimensional vector in this context can be rather tricky, due to the limitations of our reality.

On the other hand, in computer science, vectors are ordered lists of numbers, or *tuples.* For example, you might model the population of two species of animals, say, foxes and rabbits, in a given area with a pair of numbers, the first representing the number of foxes, and the second representing the number of rabbits. Note that order matters; two vectors are not equal if the numbers are swapped around.

In this context, we'd be modelling the populations as a two-dimensional vector. What makes the vector two-dimensional is that the list has two elements within it.

In maths, we are much more general. A vector is anything where we have some kind of notion of adding two objects, our *vectors*, and multiplying those vectors by a number, called a *scalar.* A *vector space* is just a set whose elements are vectors.



$$\vec{v} + \vec{w} \qquad \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a+c \\ b+d \end{bmatrix}$$

$$2\vec{v} \qquad 2\begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} 2e \\ 2f \end{bmatrix}$$

But, in a sense, what makes a vector space, a vector space, are these fundamental operations, independent of how we represent the vectors themselves - it doesn't matter whether you think about vectors fundamentally being arrows, which happen to have a nice numerical representation, or a list of numbers, which happen to have a nice visual representation as arrows. The usefulness of linear algebra is less to do with specific representations of vectors, and more to do with the ability to translate between and equate these different views.

This very general view encompasses both the arrows and ordered lists, and more, but in exchange, is very abstract, and can be possibly more difficult to pick up.

For now, we will first focus on a geometric interpretation of vectors, before moving on to more abstract vector spaces.

When we say a vector, for now, picture an arrow within a coordinate system, with the tail rooted at the origin. Note that this is somewhat distinct from the physics viewpoint discussed above, as vectors in that sense aren't tied to a specific coordinate system, and are free to move about.

This specific view is very helpful as we can then use matrix algebra in our calculations, and changing to a computer science tuple view is just as easy as reading off the coordinates of the head of the vector.

1.2 Basis Vectors, Span & Linear Independence

When we write a vector as a pair of coordinates, say,

$$\mathbf{v} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

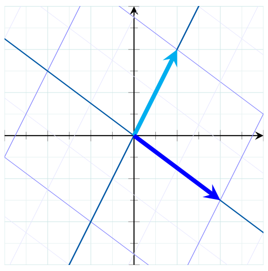
you can think of these coordinates as scalars scaling two vectors.

In the Cartesian coordinate system, there are two very special vectors we often use; the vector pointing to the right with length 1, denoted $\hat{\mathbf{\imath}}$, and the vector pointing up with length 1, denoted $\hat{\mathbf{\jmath}}$.

Thinking of the coordinates as scalars, we scale $\hat{\mathbf{\imath}}$ by 2, and $\hat{\mathbf{\jmath}}$ by 3, before adding them together to give $\mathbf{v}$, so $\mathbf{v}$ is the sum of two scaled vectors. Though this is an extremely simple example, this concept of adding two scaled vectors is worth keeping in mind, as it will soon come up, a lot. Any time we scale up vectors and add them together, it's called a *linear combination* of the vectors.

Together, $\hat{\mathbf{\imath}}$ and $\hat{\mathbf{\jmath}}$ have a special name. They are the *basis vectors* of the Cartesian coordinate system. Specifically, we call them the *canonical* or *standard* basis vectors.

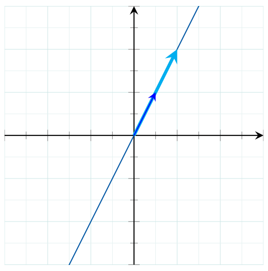
Informally, what it means to form a basis is that, when you use coordinates as scalars, the basis vectors are what the scalars act on. But this brings up a pretty interesting question. What if we picked other basis vectors?



Think about all the arrows you can get by picking two scalars, using them to scale these two arrows, then adding the results together. For these particular two arrows, the answer is that you can reach every possible two-dimensional arrow. You can see this by the fact that the transformed grid covers all of the 2D plane.

A new pair of basis vectors like this also gives us a valid way to translate between pairs of numbers and arrows in the plane. But notice that this translation is different from the canonical basis. [1,1] in this new basis certainly points to a different place than in the canonical basis. We will go into more detail later on, on how coordinates in different bases are related, but for now, just appreciate that any time we describe vectors numerically, it depends on some implicit arbitrary choice of basis vectors.

Now, if we allow the scalars to vary through all possible pairs of values, considering the linear combination given by each pair, we have three possible situations. For most pairs of vectors, we can reach every point in the plane, like in the example above. But if your two vectors line up and are parallel, then the resulting vector is also forced onto the line passing through the origin, parallel to the vectors.



Compared to the previous case, here, the transformed grid is compressed onto a single line.

Additionally, if both vectors are the zero vector, you're just stuck on the origin. The set of all possible vectors you can reach with a linear combination of a set of vectors is the *span* of the vectors. So, we can say that the span of the first pair of vectors above is the entire Cartesian plane (or equivalently, we say that the Cartesian plane is *spanned by* those two vectors, or that those two vectors form a *spanning set* of the Cartesian plane), while the span of the second pair of vectors is just a line, and the span of two zero vectors is just the single point on the origin.

In this second case, we note that one of the vectors is somewhat redundant. We can still access the full line, just using one of the vectors. In this case, we say that the vectors are *linearly dependent* - one of the vectors in the set can be expressed as a linear combination of the others, since it already lies within the span of the others.

Conversely, if each new vector adds a new dimension to the span, we say that the vectors are *linearly independent.*

Symbolically, we say that a set of vectors, $\mathbf{v}_1$,$\mathbf{v}_2$,$\mathbf{v}_3 \cdots$,$\mathbf{v}_n$ are linearly independent if the equation,

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_n\mathbf{v}_n = \mathbf{0}$$

holds only if $a_1 = a_2 = \cdots = a_n = 0$. In other words, if you can find a way to add up scaled versions of your vectors to get back to the origin, they are not linearly independent.

Now, we can more formally define a basis of a vector space as a set of linearly independent vectors that span the space, and the *dimension* of a vector space is the number of vectors in its basis.

If a linearly independent set of vectors span a space, then every vector in that space can be written as a unique linear combination of those vectors. If this spanning set is not linearly independent, then this linear combination representation of vectors will not be unique (which is why such a set is not usable as a basis - coordinates are not unique). Any two bases of the same vector space contain the same number of vectors. (Take a moment to think about why these properties are true, given the definitions we have just seen.)

2 Linear Transformations

2.1 Transformations as Matrix-Vector Multiplication

As the name suggests, in linear algebra, we only consider transformations that are linear.

Let V and W be vector spaces over a field, K (we will discuss what a field is later). A function $f : V \to W$, is linear if for any two vectors, $\mathbf{u}$,$\mathbf{v} \in V$, and any scalar, $c \in K$,

- $f(\mathbf{u} + \mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$ (*additivity* or *operation of vector addition*)
- $f(c\mathbf{u}) = cf(\mathbf{u})$ (*degree 1 homogeneity* or *operation of scalar multiplication*)

In other words, it does not matter whether the linear map is applied before or after the operations of vector addition and scalar multiplication. In particular, linear maps preserves linear combinations. Geometrically, a transformation is linear if the origin is fixed in place, and all lines remain lines under the transformation.

Although this is a rather restrictive condition, there are still a vast range of linear transformations. So, how do we represent these transformations numerically? Given a pair of numbers - a point, a coordinate - how do we find the image of that pair under any given transformation?

Looking back at the definition of a linear transformation, it doesn't matter whether we apply the map before or after the operations of vector addition and scalar multiplication, and, as discussed earlier, every vector can be seen as scaling and adding up the basis vectors - so, if we keep track of where the basis vectors are mapped under the transformation, everything else immediately follows on.

For example, if we know that,

$$\hat{\mathbf{i}} \mapsto \begin{bmatrix} 1 \\ -3 \end{bmatrix} \qquad \hat{\mathbf{j}} \mapsto \begin{bmatrix} -2 \\ 4 \end{bmatrix}$$

then we can easily tell where any arbitrary vector,

$$\mathbf{v} = \begin{bmatrix} x \\ y \end{bmatrix}$$

is mapped, by using the linear properties of these maps and breaking it down into its constituent parts, $\mathbf{v} = x\hat{\mathbf{\imath}} = y\hat{\mathbf{\jmath}}$, so,

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto x \begin{bmatrix} 1 \\ -3 \end{bmatrix} + y \begin{bmatrix} -2 \\ 4 \end{bmatrix} = \begin{bmatrix} 1x - 2y \\ -3x + 4y \end{bmatrix}$$

In doing so, we see that every two-dimensional linear map is completely determined by just 4 numbers - the coordinates of the image of $\hat{\mathbf{\imath}}$ and $\hat{\mathbf{\jmath}}$. Or more generally, the coordinates of the image of the basis vectors of the relevant space. But sticking with $\hat{\mathbf{\imath}}$ and $\hat{\mathbf{\jmath}}$ for now, we often like to package these coordinates into an array of numbers - a *matrix*.

We do this in such a way that the first column contains the coordinates of where $\hat{\mathbf{\imath}}$ lands, and the second, the coordinates for $\hat{\mathbf{\jmath}}$.

$$\begin{bmatrix} 1 & -2 \\ -3 & 4 \end{bmatrix}$$
$$\underbrace{}_{\hat{\mathbf{\imath}}} \quad \underbrace{}_{\hat{\mathbf{\jmath}}}$$

If you have the matrix for some linear transformation, and you want to know the image of any given vector, you take the coordinates of that vector, multiply them by the respective columns of the matrix, and sum the results. In other words, we are adding up the the scaled versions of the new basis vectors.

For an arbitrary matrix and vector,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \begin{bmatrix} x \\ y \end{bmatrix}$$

the image of the vector is given by,

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto x \begin{bmatrix} a \\ c \end{bmatrix} + y \begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

Since the matrix really represents a linear map - a kind of function - let's write it to the left of the vector like we normally do with functions, and give the vector as the function variable.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

But the brackets are somewhat clumsy, so we often drop them from this expression, and read the function as multiplication,

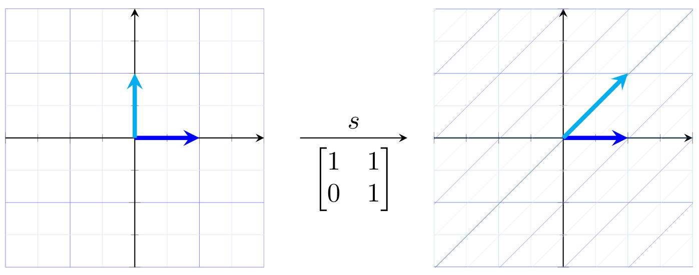
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

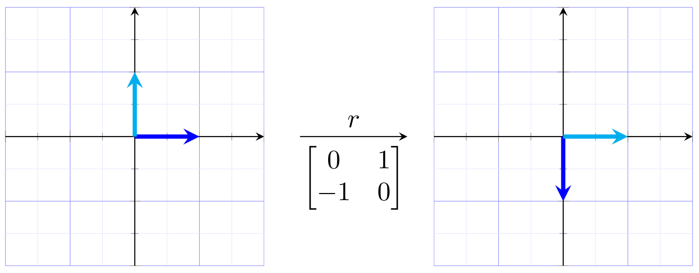
and, we've just discovered matrix-vector multiplication. If you've ever wondered why matrix-vector multiplication is what it is, this is why: it stems from linear transformations being applied to vectors.

2.2 Composition as Matrix-Matrix Multiplication

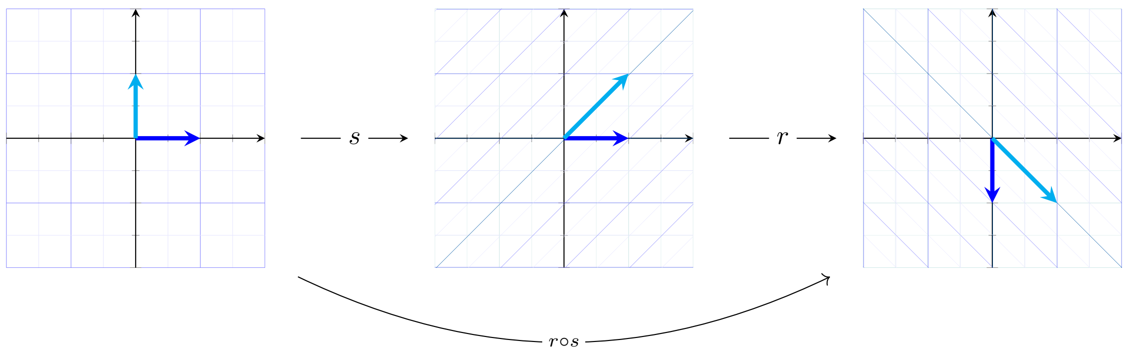
Now, we often don't study transformations in isolation: what happens if we apply two transformations to a vector, one after another?

For example, consider the transformations given by shearing parallel to the horizontal axis, and rotating by 90° clockwise.





Because of linearity, the overall effect of applying the shear then rotation is another linear transformation, distinct from both the shear and rotation alone. The new transformation is the *composition* of the two original transformations.



(We read right to left for composition. This notation stems from function notation; $(r \circ s)(\hat{\mathbf{\imath}}) = r(s(\hat{\mathbf{\imath}}))$, so we apply s first.)

Now, being a linear transformation, this composition also has a matrix representation. Above, we see that,

$$\hat{\mathbf{\imath}} \mapsto \begin{bmatrix} 0 \\ -1 \end{bmatrix} \qquad \hat{\mathbf{\jmath}} \mapsto \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{1}$$

so the composition matrix is given by,

$$\begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$$

This matrix gives the effect of shearing, then rotating, in a single transformation - one action, instead of two successive ones.

We can otherwise write the composition out in terms of the original transformations by multiplying a vector on the left by the shear, to give the image under the shear, then multiplying again by the rotation, to apply it after.

$$\underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\text{Rotation}} \left(\underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{\text{Shear}} \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) \right)$$

Given our definition of matrix-vector multiplication, this is exactly what it means to apply linear transformations as matrices to a vector.

But, given that this pair of transformations has the same overall effect as the composition matrix on any vector, it seems sensible to write

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$$

More generally, two arbitrary transformation matrices will give another transformation matrix. You probably know how matrix multiplication is defined, but put that knowledge aside for a second, and we will rederive that definition.

$$\underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_{M_2} \underbrace{\begin{bmatrix} A & B \\ C & D \end{bmatrix}}_{M_1} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$$

To figure out the overall matrix, we need to follow where $\hat{\mathbf{i}}$ goes. By how we construct matrices in the first place, the image of $\hat{\mathbf{i}}$ is just the first column of M_1. To see where that column is mapped, we then multiply that column by M_2:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} A \\ C \end{bmatrix}$$

Using our definition of matrix-vector multiplication we defined earlier, this gives

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} A \\ C \end{bmatrix} = A \begin{bmatrix} a \\ c \end{bmatrix} + C \begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} Aa + Cb \\ Ac + Cd \end{bmatrix}$$

which is the first column of the composition matrix. Similarly, for $\hat{\mathbf{j}}$, we have,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} B \\ D \end{bmatrix} = B \begin{bmatrix} a \\ c \end{bmatrix} + D \begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} Ba + Db \\ Bc + Dd \end{bmatrix}$$

so the composition matrix is,

$$\begin{bmatrix} Aa + Cb & Ba + Db \\ Ac + Cd & Bc + Dd \end{bmatrix}$$

This is where the arbitrary-feeling “rows into columns” definition matrix multiplication you learned at A-level actually comes from. It’s just how the numbers work out when we compose linear transformations together.

Furthermore, seeing matrix multiplication as composition of transformations makes the various properties of matrix multiplication much easier to understand.

For example, rotating then shearing, and, shearing then rotating, clearly give different results, so matrix multiplication is not commutative. This is a trivial property you can verify in your head, without having to compute anything at all.

Similarly, matrix multiplication is clearly assocative: applying transformation A, then (B then C) is clearly the same thing as applying transformation (A then B), then C. There’s nothing to prove here; it’s the same three transformations being applied in the same order both ways.

Trying to prove these properties symbolically is a nightmare, but, as transformation compositions, they’re trivial. Not only are these valid proofs, they’re good intuitive explanations as to why these properties should be true.

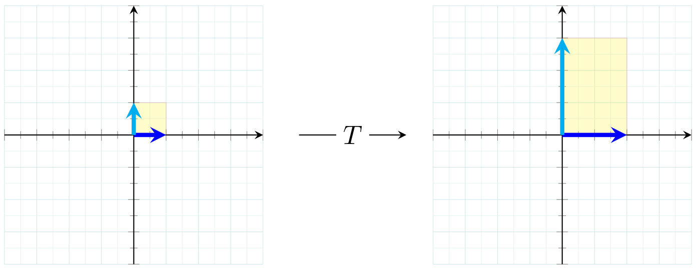
2.3 The Determinant

Consider the transformation given by the matrix,

$$\mathbf{T} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

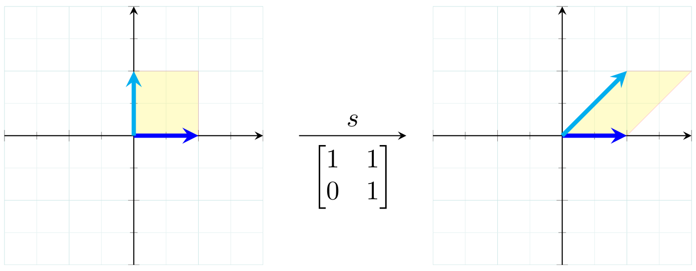
and look at how it transforms the unit square in the first quadrant (the square with sides $\hat{\imath}$ and $\hat{\jmath}$).

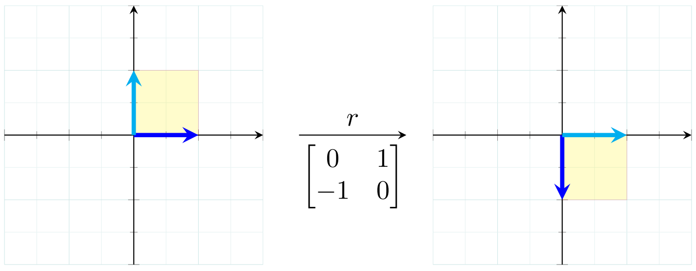
Following $\hat{\imath}$ and $\hat{\jmath}$, we find that this square is transformed into a 2 by 3 rectangle.



As the square started with an area of 1, and the resulting rectangle has area 6, this transformation has scaled the area of the square by a factor of 6.

But not all transformations will scale this square. Of course, the identity transformation trivially leaves the square unchanged, but more interestingly, the two transformations we explored in the previous section also do not affect the area of this square:





Although the shapes themselves are distorted (possibly moreso in the shear than the rotation), these transformations seem to leave areas unchanged, at least, in the case of the unit square.

However, because the transformation is linear, these transformation scale the area of *any* shape in the 2D plane by the same factor, and not just the unit square.

Recalling the geometric interpretation of a linear transformation, the origin is fixed in place, and all lines remain lines - so any square that lies within the grid containing the axes is transformed similarly to the unit square, and we can approximate any arbitrary shape that isn't a grid square as closely as we'd like with smaller and smaller squares, each of which are scaled by this same factor.

So, if we know how much the area of the unit square changes, we know how any other shape changes under that transformation.

This scaling factor is called the *determinant* of the transformation, and can variously be written as,

$$\det \mathbf{T} = \det(\mathbf{T}) = |\mathbf{T}| = \det \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} = \begin{vmatrix} 2 & 0 \\ 0 & 3 \end{vmatrix} = 6$$

This idea extends to three dimensions with scaling volume, and in arbitrary dimensions with scaling something called "*measure*" (area and volume are the specific 2D and 3D variants of measure).

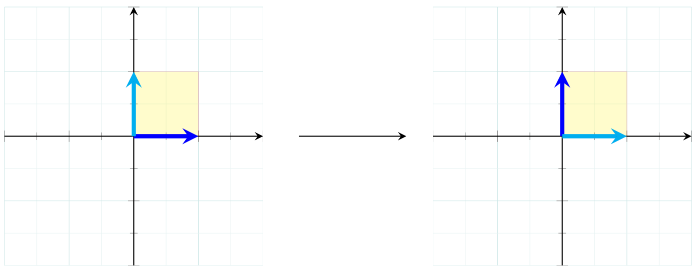
Because the shear and rotation leave areas unchanged, we also have $\det(s) = 1$ and $\det(r) = 1$.

Determinants don't have to be increases in areas either - the transformation given by,

$$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

makes areas smaller by a factor of $\frac{1}{2}$, and thus has a determinant of $\frac{1}{2}$.

The full definition of the determinant actually allows for negative values, and it doesn't really make sense to scale an area by a negative amount. This has to do with something called *orientation*.



This transformation is a reflection in the line $y = x$. You can somewhat intuitively see that space is "flipped" in some way under this transformation. Before the transformation, if you stand at the origin, facing in the direction of $\hat{\mathbf{\imath}}$, then $\hat{\mathbf{\jmath}}$ is to your left, but after the transformation, $\hat{\mathbf{\jmath}}$ is now to your right. If this is the case, we say that the orientation of space has been *inverted*.

You can similarly check the orientation of a space in 2D and 3D using a handedness rule, like with fields in physics, but, especially for arbitrary dimensions, it's generally easier to just check if the determinant is negative.

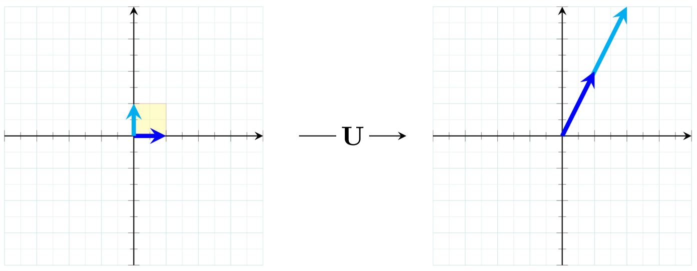
So, more properly, the *magnitude* of the determinant tells us the scaling factor of the transformation, and the *sign* tells us whether the transformation inverts the orientation of space. With this in mind, can you explain why $\det(\mathbf{A}) \det(\mathbf{B}) = \det(\mathbf{AB})$?

Now, what happens if the determinant is zero?

If we look at the matrix,

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

we notice that the image of $\hat{\mathbf{\jmath}}$ is just $\hat{\mathbf{\imath}}$ scaled by a factor of 2, so they are mapped on to the same line.



In other words, the images of $\mathbf{\hat{\imath}}$ and $\mathbf{\hat{\jmath}}$ are not linearly independent.

Because the image of a vector is given by the sum of scaled versions of $\mathbf{\hat{\imath}}$ and $\mathbf{\hat{\jmath}}$, it's clear that the image of any arbitrary vector also ends up stuck on the one-dimensional line spanned by $\mathbf{\hat{\imath}}$ and $\mathbf{\hat{\jmath}}$. In other words, the transformation compresses down all of 2D space onto a 1D line.

The square clearly now has 0 area, so the determinant of this matrix is 0. In general, the determinant of a matrix is zero if the image vectors (of your basis vectors) are linearly dependent.

We can also have another case, as given by the zero matrix,

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

This matrix compresses all of 2D space on to the origin, and also has zero determinant. We call matrices with zero determinant *singular* (and *nonsingular*, *invertible* or *nondegenerate* otherwise).

But notice the distinction between these two cases: one matrix returns a 1D line, and the other, just a single point. These are clearly very different from each other, but both fall under the bracket of "zero determinant".

2.4 Column Space and Rank

We have some terminology for this: when the output of a transformation is a line (1 dimensional), the *rank* of the matrix is 1. If all the output vectors form some two-dimensional plane, then the rank of the matrix is 2. We often call this output space the *image*, or the *column space*. This latter name comes from the columns of the matrix being the image of the basis vectors - so the space of all possible vector outputs of the matrix is just the space spanned by the columns (We similarly define the *row space* of a matrix to be the space spanned by its rows, but this is rarely used).

Note that the image of a matrix is a space (the space spanned by the transformed basis vectors), and not a matrix, while the image of a vector is another vector, and not any kind of space.

In general, the rank of a matrix is the number of dimensions in the column space

So, for the matrices above, we have,

$$\text{rank}\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = 1, \text{ and rank}\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = 0$$

For the 2×2 matrices we've been discussing, rank 2 is the best we can have. There's no way we can map a pair of vectors to span all of 3D space. We simply do not have enough basis vectors.

That isn't to say that the codomain of a linear transformation can't be of higher dimension than the codomain, just that the *image* is at most the dimension of the domain. For example, $T : \mathbf{R}^2 \to \mathbf{R}^3$, $(x,y) \mapsto (x,y,0)$ maps the 2D plane into 3D space, but notice that the image is still just a plane sitting within that 3D space.

When the rank is as high as possible, we say that the matrix is *full rank*. For a 3×3 matrix, we need rank 3 for the matrix to be full rank.

2.5 Null Space and Nullity

Note that the zero vector is always within the column space, as linear transformations must keep the origin fixed in place. In particular, for a full rank transformation, the only vector that is mapped to the origin, is the zero vector itself. But for transformations that aren't full rank - transformations that compress space down at least 1 dimension - there will be many more vectors that are mapped to zero.

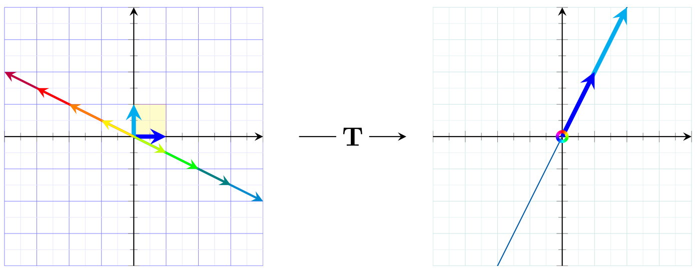
For the matrix we saw earlier, **U**, the vector,

$$\mathbf{v} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

is mapped to the origin. You may verify this yourself. But, because images are the sum of scaled basis vectors, we can actually deduce that all vectors that a multiple of this vector will also map to the origin.

$$\mathbf{v} = t\begin{bmatrix} -2 \\ 1 \end{bmatrix}, t \in \mathbb{R}$$

So, we have an entire line of vectors that are mapped to the origin:



The multicoloured vectors are all mapped to the origin, and every other vector will map to somewhere on the line spanned by the transformed basis vectors.

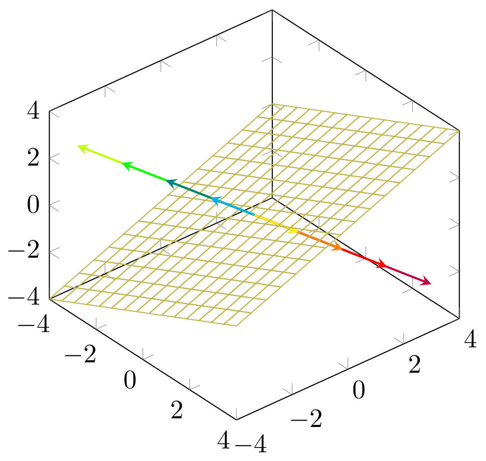
Conversely, the zero matrix maps every vector in the plane to the origin (and the column space would just be the origin, in this case).

This set of vectors that gets mapped to the origin is called the *null space* or *kernel* of the transformation. The dimension of this space is called the *nullity* of the transformation.

For **T**, since we have a line that is mapped to zero, the null space of **T** is that line, and the nullity of **T** is 1. For the zero matrix, the null space is the entire 2D plane, so the nullity is 2.

For any 3×3 matrix that maps 3D space to a plane, there will be a line of vectors that are mapped to the origin. Similarly, if 3D space is mapped on to a line, there will be an entire plane of vectors that are mapped to the origin.

Notice that these lines or planes (or volumes/spaces in higher dimensions) will always contain the origin, due to the property discussed above.



You can view nullity as the number of dimensions that are “lost” or “compressed” under a transformation, which leads us to the *rank-nullity theorem*. This theorem effectively states that the sum of the rank and nullity of a transformation is equal to the dimension of the space you are working in.

We will state this theorem more formally once we have defined what spaces even are.

2.6 Computational Skills

So, we haven’t actually discussed much computation yet. This is predominantly because there is actually very little to do for this module. The challenge is in figuring out what it is the question wants you to do, which is why so much of this is focussed on intuition and geometric understanding.

Most of the computational skills you will need are collected at the end, in the condensed summary. However, we need to cover a little bit now, before moving on to more abstract vector spaces.

2.6.1 Elementary Matrix Operations

There are three types of *row operations* we can perform on a matrix:

- Row Switching - swapping two rows;
- Row Scaling - multiplying every element in a row by a non-zero constant;
- Row Addition - replacing a row with the sum of that row and the multiple of another.

Applying a row operation to an identity matrix, then left multiplying by this new matrix is equivalent to performing the row operation on that matrix. A matrix that differs from the identity matrix by a single row operation is an *elementary matrix*.

Column operations are defined similarly, but far more rarely used in this course.

Row operations preserve row space, but not column space. They do, however, preserve the linear independence relationships between columns.

Column operations behave the exact same with with “row” and “column” in the previous paragraph switched.

2.6.2 Row Reduction

Using row operations, we can transform matrices into other forms. Some of these forms are particularly useful, and are named.

A matrix is in *row echelon form* if,

- All zero-rows are below all non-zero rows;
- The *pivot* for every non-zero row is strictly to the right of the pivot of the row above.

where a pivot is the first non-zero element of a row.

Furthermore, a matrix is in *reduced row echelon form* if,

- It is in row echelon form;
- Every pivot element is 1;
- The elements above each pivot element are 0.

Example.

$$\begin{bmatrix} 1 & 3 & 6 & 7 & 0 & 5 \\ 0 & 2 & 0 & 2 & 0 & 3 \\ 0 & 0 & 3 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

is in row echelon form, but *not* reduced row echelon form, as the pivot elements in the second, third and fourth row, have non-zero elements above them, and the pivots are not 1.

$$\begin{bmatrix} 1 & 0 & 9 & 8 & 0 & 7 \\ 0 & 1 & 3 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

is in row reduced echelon form.

The number of pivots in the row reduced matrix is the rank of the matrix. The nullity is the number of columns, minus the rank.

The row reduced echelon form of a matrix is unique.

Applying both row and column reduction to a matrix brings the matrix into *Smith normal form*, which basically looks like an identity matrix in the top left corner, with zero everywhere else.

Because row operations preserve the linear independence relations between columns, we can find the basis for the image of a matrix by row reducing the matrix, and finding the pivot columns. The vectors formed from the columns that correspond to the pivots in the row reduced matrix form a basis of the image of the matrix

Example. Find a basis for the image of,

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

A row reduces to,

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

The first and second columns are pivot columns, so the first and second columns of the original matrix,

$$\begin{bmatrix} 1 \\ 5 \\ 9 \\ 13 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \\ 10 \\ 14 \end{bmatrix}$$

form a basis of the image of $\mathbf{A}$. Because there are only two pivot columns, we only have two basis vectors, so we know the image of $\mathbf{A}$ is a 2D plane.

Because the image is the same thing as the column space - the space spanned by the columns of the matrix - if you were asked to find a linearly independent set from a set of vectors, you would just augment the vectors together into a matrix, then perform the same procedure above.

We can also find a basis of the kernel of a matrix through row reduction.

Example. Find a basis for the kernel of $\mathbf{A}$, from the above example.

We once again row reduce the matrix, but now multiply it by an arbitrary vector, and set it equal to the zero vector.

$$\begin{bmatrix} 1 & 0 & -1 & -2 \\ 0 & 1 & 1 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Now, we rewrite the arbitrary vector in terms of the variables corresponding to non-pivot columns, using the information from the matrix. For example, we know that $a = c + 2d$, so 1 and 2 are the first variables in the two vectors.

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = c \begin{bmatrix} 1 \\ -2 \\ 1 \\ 0 \end{bmatrix} + d \begin{bmatrix} 2 \\ -3 \\ 0 \\ 1 \end{bmatrix}$$

The two vectors on the right form a basis of the kernel of $\mathbf{A}$.

Once you are more comfortable with this, you can skip out writing the variables, and just read the numbers off of the matrix (but take care with signs!).

We can also extend a linearly independent set into a basis. Augment the given vectors together, then augment on any basis that spans the desired space. In practice, you can just use the identity matrix, as it is generally the easiest to work with.

Example. The vectors

$$\begin{bmatrix} 1 \\ 5 \\ 9 \\ 13 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \\ 10 \\ 14 \end{bmatrix}$$

span $\mathbb{R}^2$, as found above. Extend this set of vectors to a basis of $\mathbb{R}^4$.

Augment the vectors together, along with $\mathbf{I}_4$:

$$\begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 \\ 5 & 6 & 0 & 1 & 0 & 0 \\ 9 & 10 & 0 & 0 & 1 & 0 \\ 13 & 14 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and row reduce,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & -\frac{7}{2} & \frac{5}{2} \\ 0 & 1 & 0 & 0 & \frac{13}{4} & -\frac{9}{4} \\ 0 & 0 & 1 & 0 & -3 & 2 \\ 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

The first four columns are pivot columns, so the first four columns of the original matrix,

$$\begin{bmatrix}1\\5\\9\\13\end{bmatrix}, \begin{bmatrix}2\\6\\10\\14\end{bmatrix}, \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$$

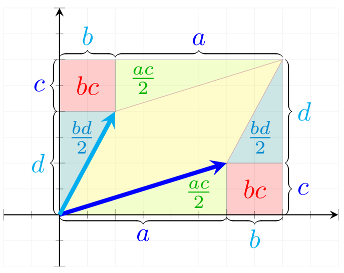
form a basis of $\mathbb{R}^4$.

2.6.3 Determinants

For a 2×2 matrix,

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

the determinant is given by $\det(\mathbf{A}) = ad - bc$. This is one place where I think a geometric explanation isn't particularly helpful, but here is a diagram, if you are still interested:



$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = (a+b)(c+d) - 2\left(\frac{ac}{2}\right) - 2\left(\frac{bd}{2}\right) - 2bc = ad - bc$$

For higher dimensional matrices, there are various methods to calculate the determinant. You probably learnt Laplacian expansion to find determinants at A-level.

However, we will often use Gaussian elimination, or more generally, row reduction, to compute the determinant of larger matrices: Scaling a row by k scales the determinant by k, and swapping two rows multiplies the determinant by -1. If we row reduce our matrix to the identity matrix (which has determinant 1), we can then run the row operations in reverse and keep track of the determinant.

2.7 Systems of Linear Equations & Matrix Inverses

One reason why linear algebra is required for such a wide variety of technical disciplines, is that it allows us to solve a certain type of system of equations.

If your system of equations only involves equations which add up multiples of your variables, i.e., linear combinations of variables, then we can apply the tools of linear algebra.

We usually organise this sort of special system of equations by putting all the variables on the left, lining them up vertically, and putting all the constants on the right.

$$\begin{aligned} 4x - 5y + 7z &= 6 \\ 2x + 3y - 2z &= 2 \\ 9x - 7y + 3z &= 5 \end{aligned}$$

This might remind you of matrix-vector multiplication. And indeed, we can wrap this entire system of equations up into a single vector equation:

$$\begin{bmatrix} 4 & -5 & 7 \\ 2 & 3 & -2 \\ 9 & -7 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \\ 5 \end{bmatrix}$$

We've separated out the constant coefficients into a matrix, and packed all of the variables into a vector, and we want their matrix-vector product to be some constant vector.

We often label the matrix $\mathbf{A}$, the variable vector as $\mathbf{x}$, and the constant vector as $\mathbf{v}$.

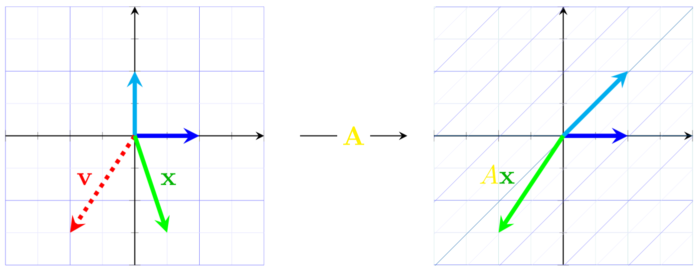
$$\mathbf{Ax} = \mathbf{v}$$

And this isn't just a notational trick to save space - we can now view this algebraic problem as a geometric one. $\mathbf{A}$ is a matrix, and therefore represents some linear transformation, despite the coefficients and source problem possibly having nothing to do with geometry. So, solving $\mathbf{Ax} = \mathbf{v}$ for $\mathbf{x}$ means we're trying to find a vector, $\mathbf{x}$, which gets mapped to $\mathbf{v}$ after applying the transformation $\mathbf{A}$.

For simplicity, suppose $\mathbf{A}$ is a 2×2 matrix.

Now, there are a couple of possible cases, depending on whether the transformation given by $\mathbf{A}$ compresses space down into a lower dimension or not. In other words, we care about whether $\mathbf{A}$ is singular or not.

If $\mathbf{A}$ is non-singular, meaning space is not compressed into a zero-area region, then there will be a single unique vector that lands on $\mathbf{v}$ under $\mathbf{A}$, and we can find it by playing the transformation backwards.



Following where $\mathbf{v}$ goes under this backwards transformation will give us $\mathbf{x}$.

Playing $\mathbf{A}$ in reverse actually gives a separate linear transformation, the *inverse* of $\mathbf{A}$, written $\mathbf{A}^{-1}$.

Here, $\mathbf{A}$ is a rightwards shear that moves $\hat{\mathbf{j}}$ one unit to the right, so $\mathbf{A}$ inverse would be a leftwards shear that moves $\hat{\mathbf{j}}$ one unit to the left. If $\mathbf{A}$ is a $90°$ clockwise rotation, then $\mathbf{A}^{-1}$ would be a $90°$ anticlockwise rotation.

In general, $\mathbf{A}^{-1}$ is the unique transformation such that, if you apply the transformation $\mathbf{A}$, then the transformation $\mathbf{A}$ inverse, the overall effect is just the identity. Applying transformations sequentially is algebraically expressed as matrix multiplication, so another way to describe this property, is that $\mathbf{A}^{-1}$ is the unique matrix such that $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$.

If you can find this inverse matrix, we can solve the equation by multiplying both sides by the inverse:

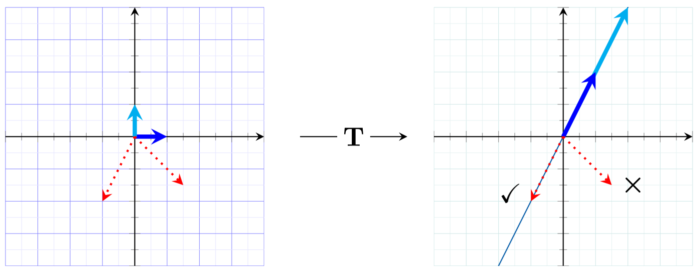
$$\begin{aligned} \mathbf{Ax} &= \mathbf{v} \\ \mathbf{A}^{-1}\mathbf{Ax} &= \mathbf{A}^{-1}\mathbf{v} \\ \mathbf{Ix} &= \mathbf{A}^{-1}\mathbf{v} \\ \mathbf{x} &= \mathbf{A}^{-1}\mathbf{v} \end{aligned}$$

This non-zero determinant case, which, for any random matrix, is almost certainly the case, corresponds with the idea that, if you have two variables and two equations, you'll almost certainly have one unique solution to the system.

This also extends to higher dimensions, when the number of equations equals the number of variables. You can translate these algebraic problems into geometric ones, finding vectors that land on other vectors under some transformation, given that the transformation associated with the coefficient matrix doesn't compress space into a lower dimension.

However, when matrix is singular, and the transformation does collapse down dimensions, then there is no inverse to find, because information is lost when compressing down space. You cannot decompress a line back into a plane - doing so would require transforming a single vector on the line into an entire line of vectors in the plane, which functions cannot do.

Solutions can still exist when the determinant is zero, but only if the constant vector just happens to be in the column space of the matrix:



Clearly, if the target vector is outside of the column space, then there's no way to get there from the basis vectors: the column space is defined as the space you can reach with linear combinations of the basis vectors. Similarly, if the target vector lies inside the column space, it's clearly reachable.

There are several ways to find the inverse of any given matrix. One which you may have learned at A-level is by using the determinant and matrix of cofactors. You may also have learned Cramer's rule, or the Cayley-Hamilton theorem, which each can also be used to (perhaps somewhat inefficiently, depending on the matrix) find the inverse of a matrix.

Here, we will often use Gaussian elimination, or row reduction, which is much faster*, especially on larger matrices.

If we have a matrix we wish to find the inverse of, we first append or *augment* the matrix with the identity matrix on the right, then row reduce the resulting rectangular matrix. The inverse matrix will then be on the right hand side.

Example. Find the inverse of:

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

First, we augment the matrix with the identity. For clarity, a separating line showing where the augmentation happened is included.

$$\left[\begin{array}{ccc|ccc} 1 & 3 & 1 & 1 & 0 & 0 \\ 0 & 4 & 1 & 0 & 1 & 0 \\ 2 & -1 & 0 & 0 & 0 & 1 \end{array}\right]$$

Then row reduce, until the identity is now on the left.

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -1 & 1 & 1 \\ 0 & 1 & 0 & -2 & 2 & 1 \\ 0 & 0 & 1 & 8 & -7 & -4 \end{array}\right] \quad \underbrace{\hphantom{-1 \quad 1 \quad 1}}_{\mathbf{A}^{-1}}$$

The inverse is then on the right.

*For an $n \times n$ matrix, Gaussian elimination takes about $O(n^3)$ time (assuming multiplication and addition takes constant time, which, they don't, as the intermediate matrix entries tend to grow exponentially in size). The cofactor matrix method requires finding the determinants of n^2 distinct $(n-1) \times (n-1)$ matrices, and finding determinants is superpolynomial in complexity. Cramer's rule takes $O(n!)$ time. In any case, even with idealisations about the speed of multiplication and addition, row reduction is generally a lot faster.

This is very useful for getting inverses, but when we are solving systems of linear equations, we often don't need the inverse itself, and are just looking for the solution vector, $\mathbf{x}$.

In this case, we can similarly rewrite a system of linear equations as an *augmented matrix* by writing the coefficients of the variables into the matrix as usual, but this time, we augment on the vector containing the constants.

Row reducing the matrix, we can immediately read off what the solutions should be. This is useful where the inverse is not needed and/or is very complicated.

Example.

$$\begin{aligned} 4x - 5y + 7z &= 6 \\ 2x + 3y - 2z &= 2 \\ 9x - 7y + 3z &= 5 \end{aligned}$$

$$\left[\begin{array}{ccc|c} 4 & -5 & 7 & 6 \\ 2 & 3 & -2 & 2 \\ 9 & -7 & 3 & 5 \end{array}\right]$$

Row reducing this matrix, we have,

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & \frac{9}{11} \\ 0 & 1 & 0 & \frac{8}{11} \\ 0 & 0 & 1 & \frac{10}{11} \end{array}\right]$$

so $x = \frac{9}{11}$, $y = \frac{8}{11}$, and $z = \frac{10}{11}$ are solutions.

You can try finding the inverse of the matrix above as an exercise, but the entries will not be very nice to work with (they are all rational, but with very large denominators).

This method also allows us to determine whether solutions exist for any given system of linear equations:

Example.

$$\begin{aligned} 3x + 6y - 6z &= -6 \\ -6x + 3y + 3z &= 2 \\ -3x - y + 3z &= -2 \end{aligned}$$

$$\left[\begin{array}{ccc|c} 3 & 6 & -6 & -6 \\ -6 & 3 & 3 & 2 \\ -3 & -1 & 3 & -2 \end{array}\right]$$

which row reduces to,

$$\left[\begin{array}{ccc|c} 1 & 0 & -\frac{4}{5} & 0 \\ 0 & 1 & -\frac{3}{5} & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

but the last line implies that $0 = 1$, so we know the system is inconsistent and has no solutions.

We can say that the coefficient matrix is of lower rank than the augmented matrix, so the system is inconsistent and has no solutions.

3 Scalars & Fields

So far, we've been saying that we can multiply or scale vectors by certain numbers called scalars. But what numbers can these scalars actually be? So far, we've mostly been using the integers or rationals, but, we can pick other numbers too.

In general, for any given vector space, the scalars must all come from a *field*.

If you have taken MA136, or are otherwise comfortable with group theory and binary operations, the following section should be easier to remember, and you can skip the list of field axioms afterwards.

Conversely, if you do not have any background in group theory, skip this outlined section and just read the field axioms.

3.1 Fields from Groups and Rings

A *ring* is a triple, $(R, +, \times)$, where R is a set and $+$ and $\times$ are binary operations such that

- R is an abelian group under $+$;
- R is closed under $\times$;
- R contains an identity under $\times$
- $\times$ is associative on R
- $\times$ left and right distributes over $+$

We call the operation denoted by $+$ *addition*, and the operation denoted by $\times$ *multiplication* or *product*, regardless of what the operations actually are. We also call the additive identity, 0_R or the *ring zero*, as it is also the zero element for the multiplication operation. We also denote the multiplicative identity, 1_R.

Furthermore, $(R, +, \times)$ is a *commutative ring* if $\times$ is commutative on R.

Note that the "commutative" part of "commutative ring" refers to multiplication, as commutativity of addition is required regardless.

However, rings notably do **not** require multiplicative inverses.

Let R be a ring and $a,b \in R$. Then,

- $a \times 0_R = 0_R \times a = 0_R$
- $-(a \times b) = (-a) \times b = a \times (-b)$

An element, a, of a ring R is a *unit* if there exists some $b \in R$ such that $ab = ba = 1_R$. Essentially, a is a unit of R if a has a multiplicative inverse in R. In any non-zero ring, 0_R is a non-unit.

Example. In $\mathbb{R}$, $\mathbb{Q}$ and $\mathbb{C}$, every non-zero element, k, has a multiplicative inverse, $\frac{1}{k}$, so the units are the non-zero elements.

However, in $\mathbb{Z}$, $\frac{1}{k}$ is an integer only for $k = \pm 1$, so the units in $\mathbb{Z}$ are ± 1.

A *field*, $(F, +, \times)$, is a commutative ring such that every non-zero element is a unit, and $0_F \neq 1_F$.

Equivalently, $(F, +, \times)$ is a field if $(F,+)$ is an abelian group with additive identity 0_F, $(F \setminus \{0_F\},\times)$ is an abelian group with multiplicative identity 1_F, $0_F \neq 1_F$ and multiplication distributes over addition.

This $0_F \neq 1_F$ condition is called the *non-degeneracy condition*, and is basically there just to exclude the trivial set, $\{0\}$, from being a field.

3.2 Field Axioms

If S is a set, then a *binary operation* on S is an operation that takes two elements, the *operands* or *arguments* of the operation, of S, and returns another element of S as its output. That is to say, it is *closed* over S.

A *field* is a set, K, together with two elements, $0_K \neq 1_K \in K$, and two binary operations, $\cdot : K \times K \to K$

and $+ : K \times K \to K$, called *multiplication* and *addition*, respectively, that satisfies the following axioms:

(A1) $\forall a,b \in K, a + b = b + a$ (commutativity of addition);

(A2) $\forall a,b,c \in K, a + (b + c) = (a + b) + c$ (associativity of addition);

(A3) $\exists 0_K \in K$ such that $\forall a \in K, a + 0_K = 0_K + a = a$ (existence of additive identity);

(A4) $\forall a \in K, \exists(-a) \in K$ such that $a + (-a) = (-a) + a = 0_K$ (existence of additive inverses).

(M1) $\forall a,b \in K, a \cdot b = b \cdot a$ (commutativity of multiplication);

(M2) $\forall a,b,c \in K, a \cdot (b \cdot c) = (a \cdot b) \cdot c$ (associativity of multiplication);

(M3) $\exists 1_K \in G$ such that $\forall a \in K, a \times 1_K = 1_K \times a = a$ (existence of additive identity);

(M4) $\forall a \in K, \exists(a^{-1}) \in K \setminus \{0\}$ such that $a*(a^{-1}) = (a^{-1})*a = 1_K$ (existence of multiplicative inverses);

(D) $\forall a,b,c \in K, (a + b)c = ac + bc$ (distributivity of multiplication over addition);

(ND) $0_K \neq 1_K$ (non-degeneracy).

Where there is no room for confusion, we write ab for $a \cdot b$, and 0 and 1 for 0_K and 1_K, respectively. We often denote general fields with the letter, K, from the German, *Körper*, meaning "corpus" or "body", suggesting a closed entity, or otherwise with an F. $\mathbb{F}$ is reserved for a certain type of finite field.

Informally, a set, K, is a field, if addition, subtraction, multiplication, and division are all well defined and function analogously to how they do on the reals.

$\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ are all fields. If you are familiar with classical constructions, the set of numbers you can construct with a straightedge and compass also forms a field.

$\mathbb{N}$ is not a field as additive inverses do not exist. (The additive identity doesn't exist either, if you exclude 0 from the naturals.) $\mathbb{Z}$ is not a field either, as not every non-zero element has a multiplicative inverse.

However, the integers modulo p is a (finite, or *Galois*) field for any prime p, commonly denoted $\mathbb{F}_p$. In particular, $\mathbb{F}_2$ is frequently used in computer science, as many logic operations that apply to bits can be converted to operations applying to elements of $\mathbb{F}_2$ (for example, adding two elements is the same as taking the XOR of those elements, and multiplying two elements is the same as taking the AND of those elements).

4 Vector Spaces

A *vector space* over a field, K, is a set, V, along with two maps, $+ : V^2 \to V$ and $\cdot : K \times V \to V$, called *vector addition* and *scalar multiplication*, respectively, that satisfies the following axioms for all $\mathbf{u},\mathbf{v},\mathbf{w} \in V$ and $a,b \in K$:

(V1) $(V,+)$ is an abelian group.

- (A1) $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$ (commutativity of vector addition);
- (A2) $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$ (associative of vector addition);
- (A3) $\exists \mathbf{0}_V$ such that $\mathbf{v} + \mathbf{0}_V = \mathbf{0}_V + \mathbf{v} = \mathbf{v}$ (existence of vector additive identity);
- (A4) $\exists(-\mathbf{v}) \in V$ such that $\mathbf{v} + (-\mathbf{v}) = (-\mathbf{v}) + \mathbf{v} = \mathbf{0}_V$ (existence of vector addition inverses);
- (A5) $\mathbf{u} + \mathbf{v} \in V$ (closure of vector addition).

(V2) $a \cdot (\mathbf{u} + \mathbf{v}) = a \cdot \mathbf{u} + b \cdot \mathbf{v}$ (distributivity of scalar multiplication over vector addition);

(V3) $(a + b) \cdot \mathbf{v} = a \cdot \mathbf{v} + b \cdot \mathbf{v}$ (distributivity of scalar multiplication over field addition);

(V4) $(ab) \cdot \mathbf{v} = a \cdot (b\mathbf{v})$ (compatibility of scalar multiplication with field multiplication);

(V5) $1_K \cdot \mathbf{v} = \mathbf{v}$ (existence of scalar multiplicative identity).

Now we have defined vector spaces, we can more formally define linear transformations as a type of function with vector spaces as domains and codomains. We can also define the image, kernel, rank and nullity of a transformation in terms of spaces.

For a linear transformation, $T : V \to W$,

$$\begin{aligned} \operatorname{im} T &= \{T(\mathbf{v}) | \mathbf{v} \in V\} \\ \ker T &= \{\mathbf{v} \in V | T(\mathbf{v}) = \mathbf{0}_W\} \\ \operatorname{rank} T &= \dim(\operatorname{im} T) \\ \operatorname{null} T &= \dim(\ker T). \end{aligned}$$

The *rank-nullity theorem* states that the dimension of the domain of a linear transformation is the sum of its rank and nullity:

For a linear transformation, $T : V \to W$, where $\dim V$ is finite, then,

$$\operatorname{rank} T + \operatorname{null} T = \dim V$$

or,

$$\dim(\operatorname{im} T) + \dim(\ker T) = \dim(\operatorname{domain} T)$$

4.1 Subspaces

Let V be a vector space over a field, K, and let $W \subseteq V$ be a non-empty set. If W is also a vector space over K, then W is is called a *subspace* of V. If $W \neq V$, then W is a *proper* subspace. If $W = \{\mathbf{0_V}\}$, then W is the *trivial* subspace. The trivial subspace has dimension 0.

You can quickly show that W is a subspace of V by verifying that W is closed under vector addition and scalar multiplication.

Let V be a vector space over a field, K, and let W_1, W_2 be subspaces of V. Then, the intersection, $W_1 \cap W_2 = \{\mathbf{w} | \mathbf{w} \in W_1 \cap W_2\}$, and sum, $W_1 + W_2 = \{\mathbf{w_1} + \mathbf{w_2} | \mathbf{w_1} \in W_1, \mathbf{w_2} \in W_2\}$, are also subspaces.

If $U = W_1 + W_2$, then every $\mathbf{u} \in U$ can be written as $\mathbf{u} = \mathbf{w_1} + \mathbf{w_2}$, where $\mathbf{w_1} \in W_1$ and $\mathbf{w_2} \in W_2$. If this representation is unique, then we write $U = W_1 \oplus W_2$, or that U is the *direct sum* of W_1 and W_2.

Two subspaces, W_1 and W_2, of a vector space, V, are *complementary* if $W_1 \cap W_2 = \{\mathbf{0}_V\}$ and $W_1 + W_2 = V$, or equivalently, $W_1 \oplus W_2 = V$.

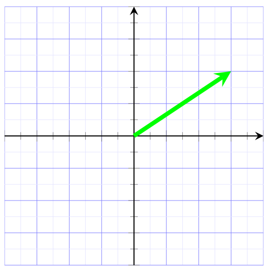
The union of two subspaces is generally not a subspace.

Let V be a vector space over a field, K, such that V is finite-dimensional, and let W_1, W_2 be subspaces of V. Then,

$$\dim W_1 + W_2 = \dim W_1 + \dim W_2 - \dim W_1 \cap W_2$$

5 Change of Basis

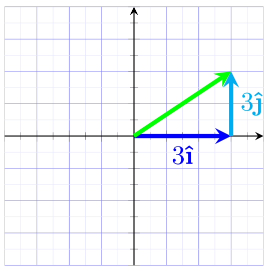
Right at the beginning, we said that assigning numbers to vectors - in the sense of arrows rooted at the origin - depends on some choice of basis vectors to provide a meaningful translation between geometry and algebra.



In our standard system, we would say that this green vector has coordinates,

$$\begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

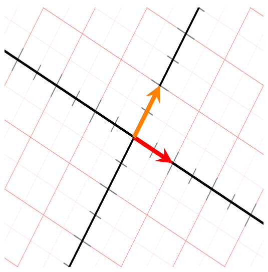
because going from its tail to its tip requires moving 3 units to the right, and 2 units up. We think of these coordinates as scalars - something that scales up a vector. In this case, we implicitly take the first coordinate to scale $\hat{\mathbf{i}}$, and the second to scale $\hat{\mathbf{j}}$, before adding up the result, with all the information about distance and direction tied up in our choice of basis vectors.



We call these ways to translate between these arrows and sets of numbers a *coordinate system*. The choice of $\hat{\mathbf{i}}$ being the target of the first scalar, and $\hat{\mathbf{j}}$ being the target of the second scalar gives us the standard Cartesian coordinate system.

But of course, other basis vectors are available.

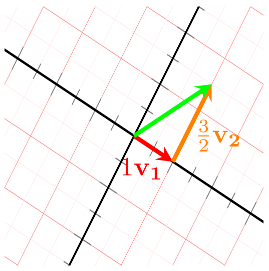
Say we have a friend, Alice, who uses a different set of basis vectors: $\mathbf{v_1}$, which points to the bottom-right and is acted upon by the first scalar of a coordinate, and$\mathbf{v_2}$, that points to the top right and is acted upon by the second scalar.



We can draw the same green vector again - the one we would describe as [3,2] - on to her grid. Alice would then describe this green vector as,

$$\begin{bmatrix} 1 \\ \frac{3}{2} \end{bmatrix}$$

What this means is that, to get to the tip of that vector using her basis vectors, is to scale up $\mathbf{v_1}$ by 1, $\mathbf{v_2}$ by $\frac{3}{2}$, then add up the results



Whenever Alice uses coordinates to describe a vector, she thinks of the first coordinate scaling $\mathbf{v_1}$, and the second, scaling $\mathbf{v_2}$, just like how we scale $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$, respectively.

We note that, although the two coordinates look different, they actually represent the same vector, just in two different coordinate systems. We're both describing the same things, but in a different language.

$$\begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{3}{2} \end{bmatrix}$$

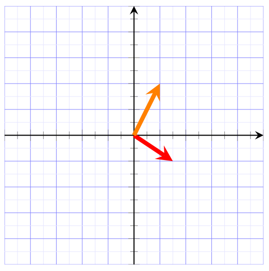
I've been showing the choices of bases using colour (and will continue doing so), but it is helpful to have notation for this as well. To do this, we give a label to each choice of basis, and subscript our vectors with that label. Often, this is done with set brackets (for example, $\{\mathbf{e}_i\}$) to indicate that the basis is a set of vectors, but here, for clarity, I will label the Cartesian coordinate system as E, and Alice's coordinate system as A.

$$\begin{bmatrix} 3 \\ 2 \end{bmatrix}_E = \begin{bmatrix} 1 \\ \frac{3}{2} \end{bmatrix}_A$$

But how did we find that second set of coordinates? More generally, how do we find the coordinates of some vector in some given different coordinate system? Well, we should first look at the basis vectors of the coordinate systems in question.

We can describe the basis vectors of the target coordinate system in terms of our standard one. In E, Alice's basis vectors are,

$$\mathbf{v_1} = \begin{bmatrix} \frac{3}{2} \\ -1 \end{bmatrix}_E \text{, and } \mathbf{v_2} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}_E$$



But it is important to note that, in her system, these vectors are

$$\mathbf{v_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}_A \text{, and } \mathbf{v_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}_A$$

since this is exactly what it means to even be a basis vector. They are what *define* the meaning of [1,0] and [0,1] in her system.

Both systems look at the same vector in space, but assign it different coordinates. The point is, the grids both of us use are just artificial constructs. Space does not intrinsically have a grid.

So, if we want to translate from our standard basis to Alice's, we want to find some kind of function that maps $[\frac{3}{2}, -1]$ in our system to [1,0] in Alice's system, and similarly, [1,2] to [0,1]. We also note that [0,0] is exactly the same in both coordinate systems: we both agree on where the origin is, since scaling any vector by 0 should always give the same result, regardless of coordinate system.

Now, on the surface, this seems rather difficult. However, it might be easier to find the translation from the Alice's basis to our standard basis, where we're looking to map [1,0] to $[\frac{3}{2}, -1]$, and [0,1] to [1,2], and you might already see where we're going with this.

If we were given some vector, say $[1, -2]$, given in Alice's coordinates, A, how would we go about translating this into our standard coordinates, E? Well, the first coordinate scales Alice's first basis vector, and similarly to the second, and we know how to express those basis vectors in our coordinate system, so we have,

$$1\begin{bmatrix}\frac{3}{2}\\-1\end{bmatrix} - 2\begin{bmatrix}1\\2\end{bmatrix} = \begin{bmatrix}-2\\4\end{bmatrix}$$

So $[1, -2]_A$ is expressed as $[-2,4]_E$ in our standard coordinate system. But...

Doesn't this look familiar?

Recalling from a long way back (§2.1), we've already done this exact same thing before! It's matrix-vector multiplication, with the matrix containing Alice's basis vectors expressed in our coordinate system.

$$1\begin{bmatrix}\frac{3}{2}\\-1\end{bmatrix} - 2\begin{bmatrix}1\\2\end{bmatrix} = \begin{bmatrix}\frac{3}{2} & 1\\-1 & 2\end{bmatrix}\begin{bmatrix}1\\-2\end{bmatrix}$$

This mapping between bases is actually a linear transformation in and of itself, and we often label its associated matrix as $\mathbf{P}$.

$$\mathbf{P} = \begin{bmatrix}\frac{3}{2} & 1\\-1 & 2\end{bmatrix}$$

In general, this matrix is given by the basis vectors of the coordinate system being converted *from*, expressed in the coordinate system being converted *to* ($\mathbf{P}$ here converts from Alice's coordinates to ours, so we use Alice's basis vectors written in our language.)

This matrix, $\mathbf{P}$, is called a *change of basis* matrix. In this case, from A to E. To convert any vector given in F to its representation in E, we left multiply by this matrix.

Since we wanted to find the coordinates for the green vector in A, given that we know its coordinates in E, we simply take the inverse, $\mathbf{P}^{-1}$, and left multiply by that instead.

$$\mathbf{P}^{-1} = \begin{bmatrix}\frac{1}{2} & -\frac{1}{4}\\\frac{1}{4} & \frac{3}{8}\end{bmatrix}$$

And you can verify yourself that this matrix, multiplied with $[3,2]_E$, gives $[\frac{3}{2},1]_A$.

From our geometric interpretation of matrix-vector multiplication from before, this is actually a very reasonable thing to do. The matrix containing the coordinates of the new basis vectors moves our basis vectors, $\hat{\imath}$ and $\hat{\jmath}$ - the things we think of as [1,0] and [0,1], - over to Alice's basis vectors - the things she thinks of as [1,0] and [0,1].

For example, if Alice was talking about a vector, say, [1,2], then multiplying [1,2] by $\mathbf{P}$ transforms our basis vectors over to Alice's, where the process of scaling then adding basis vectors by the coordinates [1,2] works in our favour, as we're effectively now working with Alice's basis vectors.

Geometrically, this matrix transforms our grid into Alice's, but numerically, it translates a vector in Alice's system into our system.

$$\mathbf{P} = \underbrace{\begin{bmatrix}a & b\\c & d\end{bmatrix}}_{\substack{\text{Alice's basis vectors}\\\text{in our coordinates}}} \qquad \mathbf{P}\overbrace{\begin{bmatrix}x_0\\y_0\end{bmatrix}}^{\substack{\text{Vector in}\\\text{Alice's coordinates}}} = \underbrace{\begin{bmatrix}x_1\\y_1\end{bmatrix}}_{\substack{\text{The same vector,}\\\text{in our coordinates}}}$$

5.1 Transformations in Different Bases

Now we can translate vectors between bases, how about transformations? If we have the 90° clockwise rotation matrix,

$$\mathbf{U} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

how would Alice represent this same transformation in her own coordinate system? To be clear, we're trying to write down a matrix that takes Alice's grid, and rotates it 90° clockwise.

The columns of the matrix encode information about where our basis vectors, $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ go, so just translating the columns into Alice's coordinates is not enough. That would just give a matrix that tells her where our basis vectors would land, written in her coordinate system.

She wants a matrix that gives where *her* basis vectors land, and it needs to describe those landing spots in her coordinate system as well.

Let's first consider what the rotation matrix does to a single specific vector, given in her coordinate system, say,

$$\begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

Since we don't know the rotation matrix in her system, let's first convert this vector into our coordinate system. We do this by using the change of basis matrix - the matrix containing her basis vectors in our coordinate system as columns.

$$\overbrace{\underbrace{\begin{bmatrix} 1 & \frac{3}{2} \\ 2 & -1 \end{bmatrix}}_{\text{Change of basis matrix, } \mathbf{P}} \begin{bmatrix} 3 \\ 4 \end{bmatrix}}^{\substack{\text{The same vector,} \\ \text{but in our language}}}$$

And now we have the vector in a form we can work with. I've left the multiplication unexpanded, but keep in mind that the whole right hand side represents a vector - the exact same vector as before, just described in our language.

Since we know the rotation matrix in our system, we can just multiply this whole thing by it:

$$\overbrace{\underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\substack{\text{Transformation matrix} \\ \text{in our language}}} \begin{bmatrix} 1 & \frac{3}{2} \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix}}^{\substack{\text{Transformed vector} \\ \text{in our language}}}$$

This tells us where the vector should go, but it's still in our language, so we convert it back into Alice's basis with the inverse change of basis matrix:

$$\overbrace{\underbrace{\begin{bmatrix} 1 & \frac{3}{2} \\ 2 & -1 \end{bmatrix}^{-1}}_{\substack{\text{Inverse change of} \\ \text{basis matrix, } \mathbf{P}^{-1}}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & \frac{3}{2} \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix}}^{\substack{\text{Transformed vector} \\ \text{in Alice's language}}}$$

And we've just figured out where some specific vector, given in Alice's language, should go, under a 90° clockwise rotation. But, since the choice of vector was arbitrary, we've found the transformation we wanted!

We apply the change of basis matrix to get the vector into a workable form, then the transformation (which we know in our language), then the inverse change of basis matrix to translate back.

$$\underbrace{\begin{bmatrix}1 & \frac{3}{2}\\ 2 & -1\end{bmatrix}^{-1}\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\begin{bmatrix}1 & \frac{3}{2}\\ 2 & -1\end{bmatrix}}_{\substack{\text{Transformation matrix}\\ \text{in Alice's language}}}\mathbf{v}$$

This composition of three matrices, together, gives the rotation matrix in Alice's coordinate system. It takes in a vector, in her language, and returns the transformed version of that vector, in her language.

We can represent all of this with the (mis)use of commutative diagrams:

$$\begin{tikzcd} V_E \arrow[r, "\mathbf{R}"] & T(V_E) \\ V_A \arrow[u, "\mathbf{P}"'] \arrow[r, "\mathbf{S}"] \arrow[ur, phantom, "\xRightarrow{T}"] & T(V_A) \arrow[u, "\mathbf{P}"'] \end{tikzcd}$$

where the transformation, $T : V \to V$ is acting on V, with the choice of basis indicated using subscripts. Note that the change of basis matrix works the same before and after the transformation - it still translates between Alice's system and ours. The transformation of space, T (notice that this is not written in bold, as it is a transformation not tied to a specific matrix), can be represented as the matrices, $\mathbf{R}$ and $\mathbf{S}$, specific to the bases E and A.

Note, since we're only dealing with one transformation that is pretty obviously an endomorphism, and only one vector space is being considered, in an exam, I would just draw the diagram above as:

$$\begin{tikzcd} E \arrow[r, "R"] & E' \\ A \arrow[u, "P"'] \arrow[r, "S"] & A' \arrow[u, "P"'] \end{tikzcd}$$

to save time. Some of the diagrams in this section will contain additional detail to aid my explanations, but by no means do you have to include every little extraneous detail when using these diagrams yourself.

Some people define the change of basis matrix to be the opposite way as is defined here, with $\mathbf{P}$ being the change of basis from E to A (so what we would call $\mathbf{P}^{-1}$), but as long as you are consistent with your arrows, the diagram makes everything clear.

On the top diagram, we want S, which directly transforms V_A to $T(V_A)$. An alternative route there, is to take P, then R, then to go along the second P arrow, but against the direction it is pointing, which indicates we should take an inverse of the matrix representing P. So, in terms of matrices, $\mathbf{S} = \mathbf{P}^{-1}\mathbf{RP}$ (reading right to left, as per function notation), matching the result from before.

Now, let's say we have another friend, Bob, who uses yet another coordinate system, distinct from both Alice's and ours. How might Alice and Bob communicate? How do we find a change of basis from Bob to Alice, and vice versa?

We can use our standard basis as an intermediary:

$$\begin{tikzcd} A \arrow[r, "P"] \arrow[d, "R"'] & B \arrow[dl, "S"] \\ E & \end{tikzcd}$$

Where R and S are the change of basis transformations from Alice's and Bob's systems to ours, as found earlier. We want P here, so we travel along R, then backwards along S, so $P = S^{-1}R$.

How would Bob give a transformation to Alice?

$$\begin{tikzcd}
A \arrow[r, "T"] \arrow[d, "R"] & A' \arrow[d, "R"'] \\
E & E' \\
B \arrow[u, "S"] \arrow[r, "U"] \arrow[uu, bend left, "P"] & B' \arrow[u, "S"'] \arrow[uu, bend right, "P"']
\end{tikzcd}$$

Following the arrows, we have,

$$U = S^{-1}RTR^{-1}S$$

or,

$$U = P^{-1}TP$$

In general, when we give a transformation between vector spaces, $T : V \to W$, we have to be careful when turning this transformation into a matrix. V and W, being abstract spaces, don't intrinsically have grids mapping them out - we have to assign basis vectors to each.

For example, let $T : \mathbb{R}^2 \to \mathbb{R}^2$ be a $90°$ clockwise rotation. The matrix for $\mathbb{R}^2_E \to \mathbb{R}^2_E$ is just the rotation matrix we're familiar with, and we found the matrix for $\mathbb{R}^2_A \to \mathbb{R}^2_A$ earlier. But a matrix giving the rotation from $\mathbb{R}^2_B \to \mathbb{R}^2_A$ is an equally valid representation of that same transformation. In the diagram above, this matrix could be given as $TR^{-1}S$, or $R^{-1}SU$.

Now, for some more terminology. Going back to the first diagram,

$$\begin{tikzcd}
V_E \arrow[r, "\mathbf{R}"] & T(V_E) \\
{} \arrow[r, Rightarrow, "T"] & {} \\
V_A \arrow[r, "\mathbf{S}"] \arrow[uu, "\mathbf{P}"] & T(V_A) \arrow[uu, "\mathbf{P}"']
\end{tikzcd}$$

Because $\mathbf{R}$ and $\mathbf{S}$ represent the same transformation, T, within the same space, V, just with respect to different bases, we call them *similar* matrices. In general, two $n \times n$ matrices, $\mathbf{A}$ and $\mathbf{B}$ are similar if you can write $\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$ for some (usually change of basis) matrix $\mathbf{P}$. Two similar matrices must be square.

More generally, for possibly rectangular $m \times n$ matrices, we have an analogous concept of *equivalence* (this equivalence is a type of equivalence relation, if you have done binary relations). Two rectangular matrices, $\mathbf{A}$ and $\mathbf{B}$ are equivalent if you can write $\mathbf{B} = \mathbf{Q}\mathbf{A}\mathbf{P}$ for two invertible matrices $\mathbf{P}$ and $\mathbf{Q}$ (the change of basis matrices for each of the pairs of coordinate systems for each space).

On a diagram, this would be,

$$\begin{tikzcd}
V_A \arrow[r, "\mathbf{R}"] \arrow[dd, "\mathbf{P}"] & W_{A'} \arrow[dd, "\mathbf{Q}"'] \\
{} \arrow[r, Rightarrow, "T"] & {} \\
V_B \arrow[r, "\mathbf{S}"] & W_{B'}
\end{tikzcd}$$

Here, V and W are vector spaces of different dimension, with subscripts indicating choice of basis. There are 4 bases in play here, as Alice and Bob each choose their own bases for both V and W. Note that, unlike the previous diagram, $\mathbf{P} \neq \mathbf{Q}$, since they are change of basis matrices in different spaces. Here,

R and S both represent the same transformation of space, then they are equivalent, as you could write $\mathbf{R} = \mathbf{Q}^{-1}\mathbf{SP}$. As a side note, to find $\mathbf{P}$, you would do the same procedure as a few diagrams ago, and use the standard basis as an intermediary, but the diagram was already getting cluttered enough, so this is left as an exercise for the reader.

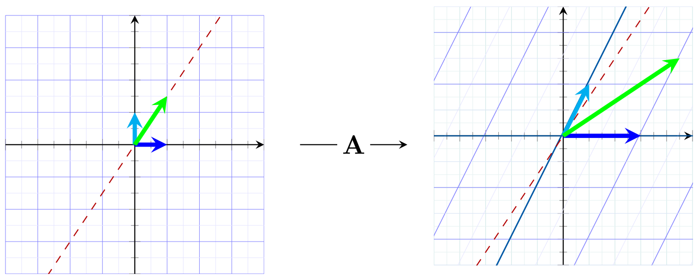
In general, similar matrices are equivalent, but equivalent matrices are not necessarily similar.

5.2 Eigenvectors

Consider the linear transformation given by the matrix,

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

and how it acts on some arbitrary vector. In particular, think about the span of that vector.



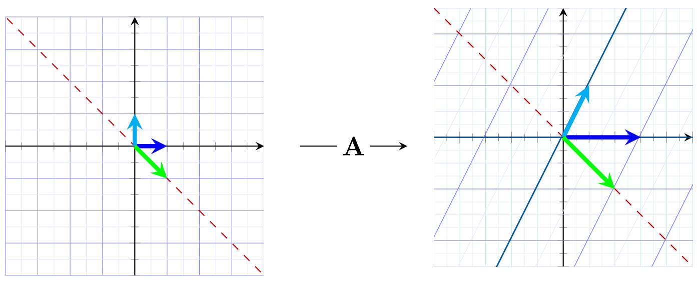
For most vectors in the plane, they get knocked off of their span during the transformation. But some special vectors do remain on their own span, meaning the transformation has no rotational effect on that vector, only scaling it by some amount.

As you might have guessed, such a vector is called an *eigenvector* of the transformation, and the amount by which it is scaled is its associated *eigenvalue*. (The prefix, -*eigen*, from German, means *own*, so an eigen-vector is an "own-vector", which makes sense, given that its image, is just itself, up to some scaling.)

For the transformation above, $\hat{\mathbf{\imath}}$ is one such vector. The span of $\hat{\mathbf{\imath}}$ is just the horizontal axis, and the image of $\hat{\mathbf{\imath}}$ clearly remains on that axis after the transformation. From the matrix, we can see that $\hat{\mathbf{\imath}}$ lands on [3,0], so it is scaled by a factor of 3. We say that [1,0] is an eigenvector of $\mathbf{A}$, with an eigenvalue of 3.

Furthermore, due to linearity, *any* vector on the horizontal axis is also similarly scaled by a factor of 3, also remaining on their own spans.

But there are more, slightly less obvious, eigenvectors to this particular transformation:



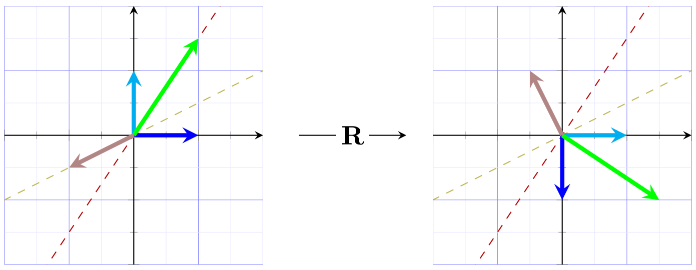
This vector, $[1, -1]$ lands on $[2, -2]$, being scaled by a factor of 2, so $[1, -1]$ is also an eigenvector of $\mathbf{A}$, with eigenvalue 2. And again, due to linearity, any vector on that line will also be an eigenvector with eigenvalue 2.

For this transformation, those are all the eigenvectors there are. Every other vector in the plane will get moved off of their spans under this transformation.

But you can also have more or fewer eigenvectors - any rotation matrix,

$$\mathbf{R} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

will move every vector off of its span.

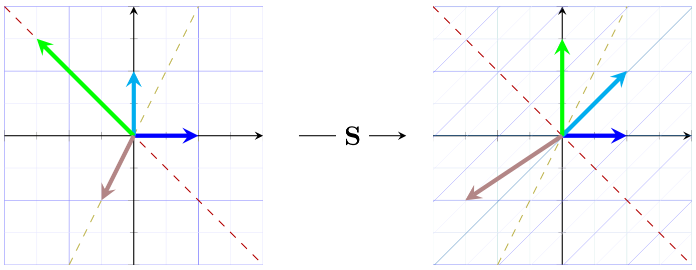


So this transformation has zero eigenvectors.

This shear,

$$\mathbf{S} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

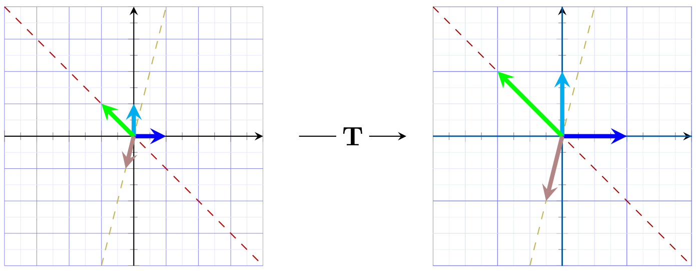
on the other hand, has every vector on the horizontal axis as an eigenvector, with eigenvalue 1, as they are unchanged by the transformation.



But every other vector is moved off of its span, so this transformation just has a single line of eigenvectors, the horizontal axis, with eigenvalue 1.

Eigenvalues don't have to be unique either - you can have multiple lines of eigenvectors with the same eigenvalue, as given by,

$$\mathbf{T} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$



This scaling matrix just stretches every vector in the plane by a factor of 2, so *every* vector is an eigenvector of this transformation, all with the same eigenvalue of 2.

By definition, the effect of a transformation, $\mathbf{A}$, on an eigenvector, $\mathbf{v}$, is just to scale it by some amount, λ, the eigenvalue. We can write this definition symbolically, as,

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

The left side is matrix-vector multiplication, while the right is scalar multiplication, so we tend to do some rearranging of this expression, by writing the right side as a matrix-vector product.

We want to scale $\mathbf{v}$ by a scalar, λ. The columns of the desired matrix need to scale each basis vector by λ, so this matrix will have λ across the diagonal, and zeros everywhere else.

$$\begin{bmatrix} \lambda & 0 & 0 & \cdots & 0 \\ 0 & \lambda & 0 & \cdots & 0 \\ 0 & 0 & \lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda \end{bmatrix}$$

Factoring out the λ, this is just the identity matrix.

$$\mathbf{A}\mathbf{v} = (\lambda\mathbf{I})\mathbf{v}$$

And now both sides are a matrix-vector product. We can then subtract the right side, and factor out the $\mathbf{v}$,

$$\begin{aligned} \mathbf{A}\mathbf{v} - (\lambda\mathbf{I})\mathbf{v} &= \mathbf{0} \\ (\mathbf{A} - \lambda\mathbf{I})\mathbf{v} &= \mathbf{0} \end{aligned}$$

The expression inside the bracket is just a matrix - the original transformation matrix, $\mathbf{A}$, but with a λ being subtracted from the diagonal, and would look something like,

$$\begin{bmatrix} 3-\lambda & 1 & 4 \\ 1 & 5-\lambda & 9 \\ 2 & 6 & 5-\lambda \end{bmatrix}$$

We're looking for a vector, $\mathbf{v}$, such that this new matrix maps $\mathbf{v}$ to the zero vector.

If $\mathbf{v} = \mathbf{0}$, then this is trivially true and isn't particularly helpful, so we're looking for non-zero solutions for $\mathbf{v}$ - a non-zero eigenvector.

Recalling terminology from earlier (§2.5), we're looking for a non-zero vector that lies in the null space of $(\mathbf{A} - \lambda\mathbf{I})$. If the null space of $(\mathbf{A} - \lambda\mathbf{I})$ is non-empty, i.e., there exists an non-zero eigenvector, it follows that $(\mathbf{A} - \lambda\mathbf{I})$ cannot be full rank, and therefore has a zero determinant (if this doesn't make immediate sense, reread the definitions of null space and rank, and take a few moments to consider what it means for a transformation to have a non-empty null space).

In other words, the only way for a non-zero vector to be mapped to the origin, is if the transformation collapses space down into a lower dimension, corresponding to a zero determinant.

So, we're trying to solve,

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

for λ. This is called the *characteristic equation* of the matrix. The left side by itself is called the *characteristic polynomial*.

For example, earlier, we had,

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

$$\begin{aligned}
\mathbf{A} - \lambda\mathbf{I} &= \begin{bmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{bmatrix} \\
\det(\mathbf{A} - \lambda\mathbf{I}) &= (3-\lambda)(2-\lambda) - (1\cdot 0) \\
\det(\mathbf{A} - \lambda\mathbf{I}) &= 0 \\
0 &= (3-\lambda)(2-\lambda) \\
\lambda &= 3{,}2
\end{aligned}$$

matching the results from before. Then, to find the actual eigenvectors, multiply the modified matrix by an arbitrary vector, and set it equal to 0.

For $\lambda = 2$, we have

$$\begin{aligned}
\begin{bmatrix} 3-2 & 1 \\ 0 & 2-2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
\begin{bmatrix} x+y \\ 0 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
x + y &= 0 \\
y &= -x
\end{aligned}$$

So any vector of the form,

$$\begin{bmatrix} t \\ -t \end{bmatrix}$$

is an eigenvector with eigenvalue 2. We generally just pick one single eigenvector as a representative for this entire line, so letting $t = 1$, we have $[1, -1]$, as before.

For a transformation which has multiple eigenvectors with the same eigenvalue, you'll find that the simultaneous equations in the final step will have multiple solutions, corresponding to the multiple eigenvectors.

Doing the same process for the rotation matrix,

$$\begin{aligned}
\mathbf{R} - \lambda\mathbf{I} &= \begin{bmatrix} 0-\lambda & 1 \\ -1 & 0-\lambda \end{bmatrix} \\
\det(\mathbf{R} - \lambda\mathbf{I}) &= (-\lambda)(-\lambda) - (1\cdot(-1)) \\
\det(\mathbf{R} - \lambda\mathbf{I}) &= 0 \\
\lambda^2 + 1 &= 0 \\
\lambda &= \pm i
\end{aligned}$$

we find that there are no real eigenvalues for this transformation. The eigenvalues of $\pm i$ correspond to the fact that multiplying by i represents a 90° rotation in the complex plane, and the magnitude of the eigenvalues being 1 corresponds to the fact that vectors aren't scaled under this transformation. In general, imaginary components of eigenvalues correspond to some kind of rotation. We can still solve for eigenvectors, but they will have complex components.

As one very basic application of eigenvectors, if you can find an eigenvector of a 3D rotation, you've found the axis of rotation - and it's much easier to think of rotations in 3D as an angle around an axis, rather than the entire 3×3 rotation matrix.

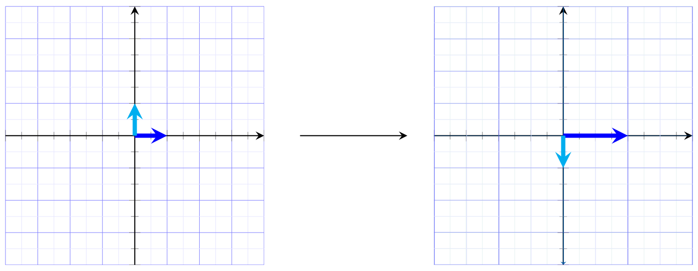
This is a common theme throughout linear algebra - with any linear transformation given as a matrix, we can interpret what it is doing by looking at its columns and seeing where the basis vectors are mapped.

But this puts a lot of emphasis on coordinate systems - another way, less dependent on coordinate systems, is to look at the eigenvectors and eigenvalues.

Two similar matrices - two matrices representing the same linear transformations, but in different coordinate systems - will have the same characteristic equation, and the same eigenvalues. Changing the coordinate system doesn't change the eigenvalues of a transformation - regardless of how you label space, eigenvectors are scaled the same way.

For another, much more general application, consider what happens if our basis vectors both happen to be eigenvectors. Let's start in the canonical coordinate system, and say we have a linear transformation such that,

$$\hat{\mathbf{i}} \mapsto \begin{bmatrix} 2 \\ 0 \end{bmatrix} = 2\hat{\mathbf{i}} \qquad \hat{\mathbf{j}} \mapsto \begin{bmatrix} 0 \\ -1 \end{bmatrix} = -1\hat{\mathbf{j}}$$



So the matrix associated with that transformation would be,

$$\begin{bmatrix} 2 & 0 \\ 0 & -1 \end{bmatrix}$$

Notice how the eigenvalues of the basis vectors lie along the diagonal of the matrix, and every other entry is zero. Any matrix with this property is called a *diagonal matrix*, and we've met one before - the identity matrix is a diagonal matrix.

The way to interpret a diagonal matrix, is that all the basis vectors are eigenvectors, with the eigenvalues written along the diagonals.

There are many reasons why diagonal matrices are much nicer to work with. One application is in taking powers of matrices, or equivalently, applying a transformation to a vector many times. Since a diagonal matrix only scales each basis vector by some eigenvalue, applying that matrix to a vector n times, just means you scale each basis vector by the eigenvalue to the power of n.

$$\begin{aligned}
\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 2x \\ 3y \end{bmatrix} \\
\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 2^2x \\ 3^2y \end{bmatrix} \\
\underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}\cdots\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}}_{100}\begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 2^{100}x \\ 3^{100}y \end{bmatrix} \\
&= \begin{bmatrix} 2^{100} & 0 \\ 0 & 3^{100} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}
\end{aligned}$$

Just looking at the transformation overall, we can write an exceedingly simple formula for the nth power of a diagonal matrix:

$$\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}^n = \begin{bmatrix} a^n & 0 \\ 0 & b^n \end{bmatrix}$$

In contrast, try calculating the 100th power of a non-diagonal matrix. There is no simple pattern to find.

For a few more nice properties of diagonal matrices, the determinant of a diagonal matrix is just the product of the diagonal. This is because each entry on the diagonal tells us how much the basis vector is scaled in that direction, so the product of all of these entries gives us how much measure is scaled overall.

Of course, all of this is only useful when the matrix we're working with is diagonal - when our basis vectors just happen to both be eigenvectors.

However, if your transformation has a lot of eigenvectors, enough so we can choose a set that spans the space the transformation is acting on, then we could use a change of basis matrix to change those eigenvectors to be our basis.

For the matrix earlier,

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

we found two eigenvectors,

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{and } \mathbf{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

with eigenvalues $\lambda_1 = 3$, and $\lambda_2 = 2$, respectively.

We use the eigenvectors as the columns of a change of basis matrix, and change the transformation matrix into our new basis, as before.

$$\begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}$$

Because we've chosen the basis vectors to be eigenvectors, we know that resulting matrix will be a diagonal matrix, with the corresponding eigenvalues along the diagonal, without even doing any calculations.

$$\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$$

This is because we're now working in a basis, where the effect of this specific transformation on these basis vectors is just to scale them by these eigenvalues.

A basis where every basis vector is an eigenvector is called an *eigenbasis*, and this process of changing a matrix to an eigenbasis is called *diagonalisation.*

If we wanted to calculate the 100th power of **A**, we could change to an eigenbasis, calculate the power there, using our simple formula, then change back.

But not every transformation admits an eigenbasis. The shear we saw earlier, for example, only has a single line of eigenvectors, which isn't enough to span all of 2D space.

6 Abstract Vector Spaces

At the very beginning of this document, we asked the question, "What are vectors?".

Is a vector fundamentally an arrow, which we can describe with coordinates, or are they fundamentally lists of numbers, which just happen to have a nice visualisation. Or are both of these views manifestations of something deeper?

Thinking of vectors as primarily being lists of numbers makes them very straightforward and intuitive. Things like four-dimensional or n-dimensional vectors are very easy to think of in this context - they're just longer lists of numbers. Working in 2 dimensions is just as easy as working in 200. Otherwise, four-dimensional space is just some strange, geometric idea that can't be easily visualised or described.

On the other hand, as you get more used to working in linear algebra - particularly with changing your basis - you'll find that many concepts are inherently to do with a space that exists independently from any choice of coordinates. Core ideas like determinants and eigenvectors don't care about the coordinate systems. The determinant tells you how much a transformation scales area, or volume, or measure, and eigenvectors are the ones which stay on their own span under a transformation. Both of these ideas are inherently spacial, and their algebraic equivalents seem completely arbitrary when seen alone.

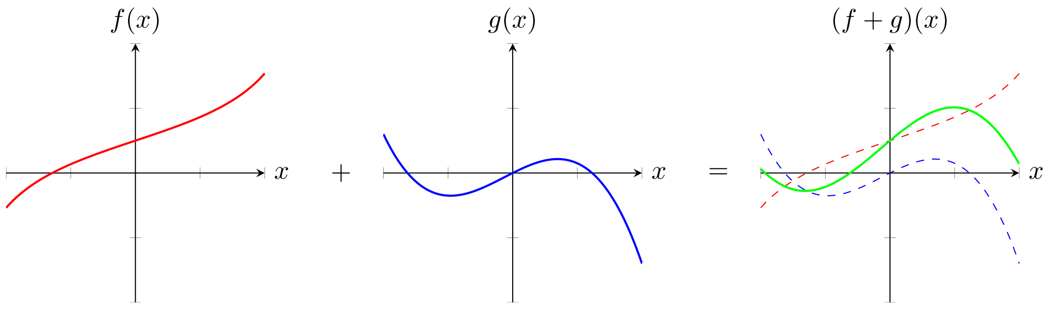
If vectors are neither arrows nor lists of numbers, and are some abstract concept to do with space, we still haven't really answered the question. We've just stated things that aren't fundamental to a vector.

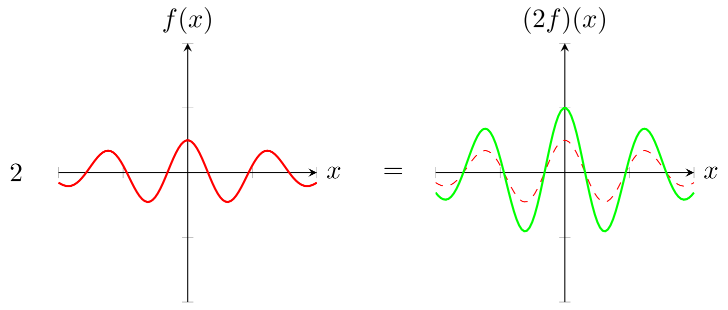
To answer this question more deeply, let's discuss something that is neither arrows, nor lists of numbers: functions.

In the same way we can add two vectors together and multiply them by a scalar,

$$\begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a+c \\ b+d \end{bmatrix}$$
$$2\begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} 2e \\ 2f \end{bmatrix}$$

we can similarly add two functions, f and g to get a new function, $(f+g)$, or multiply a function by a number to scale it, $2f(x) = (2f)(x)$:





The value of the sum function, $(f+g)$ at any given input x is the sum of the values of $f(x)$ and $g(x)$, so $(f+g)(x) = f(x) + g(x)$.

This is similar to adding vectors together, coordinate by coordinate, just that, we now have uncountably infinitely many coordinates - one for each possible input x.

For scaling a function, we just multiply each output by the scalar, just as we do for vector components, though again, here we have uncountably infinitely many coordinates.

Functions appear to be some kind of infinite-dimensional vector-ish thing. A reasonable question is, what kind of transformations of functions are there that are linear? What does it even mean for such a function to be linear?

Although functions don't look like vectors, we can still use the symbolic definition of linearity from before (§ 2.1). In the context of functions, these transformations are called *operators* instead, but they're really the same thing.

One example of an operator you'll be familiar with, is the derivative - it's something that transforms one function, into another.

If you add two functions, then take the derivative, it's the same as first taking the derivative of the two functions, then adding them:

$$L(\mathbf{u}+\mathbf{v}) = L(\mathbf{u}) + L(\mathbf{v})$$

$$\Updownarrow$$

$$\frac{d}{dx}(x^3 + x^2) = \frac{d}{dx}(x^3) + \frac{d}{dx}(x^2)$$

Similarly, scaling a function, then taking the derivative is the same as taking the derivative, then scaling the result:

$$L(c\mathbf{v}) = cL(\mathbf{v})$$

$$\Updownarrow$$

$$\frac{d}{dx}(cx^2) = c\frac{d}{dx}(x^2)$$

We can see that the differential operator is linear - in fact, the linearity requirements are exactly the sum rule and the constant factor rule from differential calculus.

One of the most important consequences of linearity, is that a linear transformation is completely described by its action on a basis. Since any vector can be expressed by scaling and adding the basis vectors in some way, finding the image of a vector under a transformation is simplified down to finding the image of the basis vectors. This is just as true for functions, as it is for arrows, or lists of numbers.

Because the differential operator is a linear transformation, we should be able to express it as a matrix. For now, let us restrict our space to the space of polynomials, so each element in our space is a polynomial with finitely many terms, but the whole space includes polynomials of arbitrarily large degree.

First, we need to pick a basis for the space of functions. Since polynomials are already written as a linear combination of powers of x, we can just choose monomials in x to be our basis.

So, the polynomial, $1 + 5x + 4x^3$ would be written as,

$$\begin{bmatrix} 1 \\ 5 \\ 0 \\ 4 \\ \vdots \end{bmatrix}$$

with infinitely many zeros following on. You can read this as 1 times the first basis function, which is just 1, plus 5 times the next basis function, x, plus 0 times x^2, 4 times x^3, plus zero times all the other basis functions.

Since polynomials only have finitely many terms, every polynomial will be represented by a finite string of numbers at the top of the vector, followed by infinitely many zeros.

In this coordinate system, the derivative is described by the infinite matrix,

$$\frac{d}{dx} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 2 & 0 & \cdots \\ 0 & 0 & 0 & 3 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

which is mostly full of zeros, but has the natural numbers running down this offset diagonal. To get a feel of why it works, cut the matrix off at a finite point, and multiply some vector by it.

This is all possible because the derivative operator is linear. If you had wanted to construct this matrix yourself, you could consider what the derivative operator does to each basis function, and put the coordinates of the results into each column.

For example, the derivative of the first basis function, 1, is 0, which has a vector representation of $[0,0,0,\cdots]$, corresponding to the first column of the matrix. Then, the derivative of the next basis function, x, is 1, which is $[1,0,0,\cdots]$, then x^2 is $2x$, which is $[0,2,0,\cdots]$.

So, it turns out that taking a derivative and matrix-vector multiplication are really members of the same family. In fact, the vast majority of concepts in linear algebra have direction analogues with respect to functions:

Linear transformations $\Leftrightarrow$ Linear operators

Dot products $\Leftrightarrow$ Inner products

Eigenvectors $\Leftrightarrow$ Eigenfunctions

There are a lot of vector-ish things in maths, all of which have their own analogues to these concepts. The answer I gave to "what is a vector" before was, anything where we have some kind of notion scaling and adding, whether that's a set of arrows in space or lists of numbers. As long as those two requiremnts hold, all of the tools of linear algebra regarding vectors - linear transformations, eigenvectors, determinants - should be able to apply.

If you're a mathematician developing new theory, you want all of your new definitions to fully apply to all of these vector-ish things, and not just some specific sub-cases. So, we use the axioms we defined before, as an interface between different vector spaces. You, as the mathematician, never have to think about all the vector spaces that could possibly exist - you just have to prove your results in terms of these axioms, and anyone who can prove that their new crazy space follows those axioms, can apply your results, even if you've never even thought about their space before. As a consequence, all of our results tend to be expressed extremely abstractly - only in terms of the axioms, rather than on a specific view, like arrows, lists of numbers, or functions.

So, the mathematician's answer to "what is a vector?" is just to ignore the question. In modern theory, it doesn't matter what the vectors are themselves - it's the fact that they obey the vector space axioms that matter.

On the topic of abstraction, what's really the difference between a linear transformation and a non-linear one? And why do we have to work with scalars from fields? Why not rings? Or groups? Or even things that aren't sets?

If you're interested in even higher level abstractions, they key phrases to search for are *category theory* and *morphisms*.

In linear algebra, we consider vector spaces - sets of vectors over a field - and linear transformations - transformations which obey certain rules - between them. In category theory, we consider categories - two "sorts" of "objects" - and morphisms - things which relate objects.

These requirements seem a lot... looser than in linear algebra, and, they are! Just like how we can apply theorems in linear algebra to a vast variety of vector spaces, theorems in category theory apply to a vast variety of categories - and the highly abstract nature of categories means that these results can be extremely general.

For example, a *homomorphism* is a map between two mathematical structures of the same type that preserves that structure. In set theory, homomorphisms are functions you are familiar with; in group theory, group homomorphisms; and linear algebra, linear transformations. If you prove a result about these in terms of categories, you've got a result that works in linear algebra, group theory, ring theory, etc. You even get morphisms between morphisms (and even morphisms between those, but I think we've gotten meta enough), which allows you to apply some results between categories, such as applying a linear algebra result to group theory.

If you are interested in any kind of abstract algebra, I highly recommend reading up on this topic. *Algebra* by Serge Lang, *Categories for the Working Mathematician* by Saunders Mac Lane, and *Categories and Sheaves* by Kashiwara and Schapira are good introductions to category theory.

7 Closing Remarks & Computational Skills

The following is a short list of computational skills which will give you the easy marks at the beginning of most questions. Beyond this, just immediately row reduce any matrix you get given before even thinking about anything else, and you've probably already done most of the question. The key to this module is having a strong geometric understanding of linear algebra, which is why this particular document is so long, relative to many of my other guides.

Prove that the vectors, v_1,$v_2, \cdots$,v_n are linearly independent:

Prove that $c_1v_1 + c_2v_2 + \cdots + c_nv_n = a$ only if $c_1 = c_2 = \cdots = c_n = 0$.

Prove that a set of vectors span a space of dimension n:

Augment the vectors together into a matrix and row reduce. If the rank of the matrix is n, then the vectors span the space.

Prove that a set of vectors are a basis of the space:

Do both of the above.

Determine the rank of a matrix:

Row reduce the matrix and count the number of pivot columns.

Determine the nullity of a matrix

Count the number of columns minus the rank, as found above.

Determine the determinant of a matrix:

Do Laplacian expansion (see A-Level), or row reduce to the identity matrix. Scaling a row by k scales the determinant by k, and swapping two rows multiplies the determinant by -1. After row reducing to the identity matrix (which has determinant 1), run your row operations in reverse and keep track of the determinant.

Find basis of image:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

$\mathbf{A}$ row reduces to,

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

The first and second columns are pivot columns, so the first and second columns of the original matrix,

$$\begin{bmatrix} 1 \\ 5 \\ 9 \\ 13 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \\ 10 \\ 14 \end{bmatrix}$$

form a basis of the image of $\mathbf{A}$. Because there are only two pivot columns, we only have two basis vectors, so we know the image of $\mathbf{A}$ is a 2D plane.

Find basis of kernel:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

A row reduces to,

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

Multiply by an arbitrary vector, and set it equal to the zero vector.

$$\begin{bmatrix} 1 & 0 & -1 & -2 \\ 0 & 1 & 1 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Now, we rewrite the arbitrary vector in terms of the variables corresponding to non-pivot columns, using the information from the matrix.

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = c \underbrace{\begin{bmatrix} 1 \\ -2 \\ 1 \\ 0 \end{bmatrix}} + d \underbrace{\begin{bmatrix} 2 \\ -3 \\ 0 \\ 1 \end{bmatrix}}$$

These are the basis vectors

The two vectors on the right form a basis of the kernel of **A**.

Once you are more comfortable with this, you can skip out writing the variables, and just read the numbers off of the matrix (but take care with signs!).

Find eigenvalues of a matrix:

Solve $\det(\mathbf{A} - \lambda\mathbf{I}) = \mathbf{0}$ for λ.

Find eigenvectors of a matrix:

Solve $(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$ for $\mathbf{v}$.

Find change of basis matrix from given basis to canonical basis:

Write matrix with given basis vectors as columns.

Find change of basis matrix from canonical basis to given basis:

Invert the above.

Find change of basis from one given basis, E_1, to another given basis, E_2 (but both of the same vector space):

Let E_0 be the canonical basis. Find $E_0 \to E_1$ as above, and let that matrix be **A**. Find $E_0 \to E_2$ as above, and let that matrix be **B**. Then, the matrix, **P**, for $E_1 \to E_2$ is given by $\mathbf{P} = \mathbf{B}\mathbf{A}^{-1}$.

$$\begin{tikzcd}
E_1 \arrow[r, "P"] & E_2 \\
E_0 \arrow[u, "A"] \arrow[ur, "B"'] &
\end{tikzcd}$$

We want P, so we travel along A backwards, then along B, so $\mathbf{P} = \mathbf{BA}^{-1}$.

<u>Find transformation T in basis $E_1 \to T(E_1)$, given matrix $A : E_0 \to T(E_0)$:</u>

$$\begin{tikzcd}
E_0 \arrow[r, "A"] & T(E_0) \\
E_1 \arrow[u, "P"'] \arrow[r, "B"] & T(E_1) \arrow[u, "P"']
\end{tikzcd}$$

We want B, so follow the path along P, then A, then P^{-1}, so $\mathbf{B} = P^{-1}\mathbf{AP}$.

<u>Find inverse of matrix:</u> Matrix of minors and cofactors, etc., as in A level, or; Augment matrix with identity and row reduce until LHS is identity. RHS will be the inverse.

<u>Extend set of vectors to basis</u> Augment vectors together into a matrix, and augment with the appropriate identity matrix. Row reduce, then note which columns are pivot columns. The columns in the original matrix which correspond to the pivot columns form a basis.