Physics at Random

Daniel Earwicker

September 5, 2024

Contents

Chapter 1

Miscellany

There are a few basic ideas that are of fundamental importance and can be summarised in a basic way quite briefly.

1.1 Linearity

Given a function $\mathbf{f}(x)$ and some constant c, we can ask if it matters whether we multiply the input x by c and then apply the function to the result, or we apply the function to x and multiply the result by c. That is, does this equation hold?

$$\mathbf{f}(cx) = c\mathbf{f}(x)$$

Or to put it another way, if we "scale up" the input, does the output scale up in proportion?

Also for two variables x and y:

$$\mathbf{f}(x + y) = \mathbf{f}(x) + \mathbf{f}(y)$$

That is, it may be that it doesn't matter whether we sum the inputs and then apply the function to the sum, or apply the function to each input and then sum the results.

If both these equations hold, we say $\mathbf{f}$ is linear.

Actually if you take the addition rule and set $y = x$:

$$\mathbf{f}(x + x) = \mathbf{f}(x) + \mathbf{f}(x)$$

or:

$$\mathbf{f}(2x) = 2\mathbf{f}(x)$$

Which is surely a huge clue about the scaling rule! Though neither is a complete statement of linearity without the other.

Sometimes these are combined into a single, albeit more confusing, requirement:

$$\mathbf{f}(ax + by) = a\mathbf{f}(x) + b\mathbf{f}(y)$$

We can generalise this concept beyond functions that act on numbers. Think of $\mathbf{f}$ as an operator. The objects it operates on can be of any type for which we can define addition and scaling (multiplication by a constant), as that's all we need to check the linearity requirement.

We can define these capabilities for vectors, matrices and indeed all tensors, so operators acting on all those things can be linear. Now, it's easy to see how this might happen, because all those things can be described by scalar components, which can themselves be added and scaled.

So let's consider something way more abstract. It's also commonplace to define addition for functions (forget about our previous use of $\mathbf{f}$):

$$\mathbf{h} = \mathbf{f} + \mathbf{g}$$

The sum of two functions is another function, one whose value is the sum of the values of the other two functions for the same input:

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

And similarly we can scale a function, to make another function:

$$\mathbf{h} = k\mathbf{f}$$

$$\mathbf{h}(x) = k\,\mathbf{f}(x)$$

If we encounter an operator $\hat{O}$ that somehow acts on a function to produce another function, we can ask if $\hat{O}$ is linear. That is:

$$\hat{O}(\mathbf{f} + \mathbf{g}) = \hat{O}(\mathbf{f}) + \hat{O}(\mathbf{g})$$

is true, as is:

$$\hat{O}(k\mathbf{f}) = k\,\hat{O}(\mathbf{f})$$

Note that an operator is not restricted to mappings that perform arithmetic on parameters. An operator may dig into the *definition* of a function and transform it through analysis (in coding terms, an operator can read the source of the input function, not merely call it.)

So an example of an operator would be differentiation. A function such as $\sin$ can be differentiated analytically and the result is $\cos$. If we differentiate $\cos$ we get $-\sin$. It doesn't matter if we:

- add the functions, then differentiate, or
- differentiate the functions, then add

Either way, we end up with $\cos - \sin$. This is true whatever functions we're adding, because differentiation works on each term individually and then adds the results.

The same goes for scaling, because when you amplify a function, you amplify the slope of the function.

> Of course, by scaling a function we mean multiplying it by a *constant*; if we multiplied $f(x)$ by another function $g(x)$, the gradient curve could end up with a wildly different shape. If we differentiate $f(x)$ and then multiply it by $g(x)$, we've skipped the differentiation of g.

So, the "differentiation operator" meets the requirements of linearity, so differentiation is linear (and intuitively as integration is the inverse operation of differentiation, it too must be linear).

Another example is the Fourier transform, $\mathcal{F}$ (§1.4). If you add two waves and take the Fourier transform of the combined wave, you get the same frequency distribution as if you took the Fourier transform of each wave separately and then added the two frequency distributions:

$$\mathcal{F}(\mathbf{g} + \mathbf{h}) = \mathcal{F}(\mathbf{g}) + \mathcal{F}(\mathbf{h})$$

And unsurprisingly, it's the same story with scaling:

$$\mathcal{F}(k\mathbf{g}) = k\,\mathcal{F}(\mathbf{g})$$

(And the same for $\mathcal{F}^{-1}$ as you'd expect.)

This next one is a little looser as an analogy. We can classify all objects in a binary way, dividing them into members and non-members of some set. Suppose we come up with a sense in which we can add two members of the set, or

scale them. Is the result always a member of the set also? If so, that's a kind of linearity.

For example, if two functions are solutions to the Schrödinger equation with some potential, they may be scaled and added to produce a third solution, so we say the Schrödinger equation is linear.

1.2 Indices

An index (plural: indices) is a subscript (and in some contexts a superscript) that stands for an integer.

$$a_n$$

This tells us that a is not just one value, but several. The n can be assumed to take a small range of values such as $1, 2, 3$, the exact size of this range depending on the situation. (In physics if we have a function of integers it's usually written as a set with an index like this, with $f(x)$ reserved for functions of continuous values.)

Instead of labelling spatial coordinates x, y, z, we can call them x_1, x_2, x_3 and avoid the need to repeat ourselves by just giving the rule for the behaviour of x_n, which is then unambiguously the same for all three dimensions of space.

Often we want to add the values:

$$x_1 + x_2 + x_3 + \cdots + x_n$$

The shorthand for this is to use the Σ symbol:

$$\sum_n x_n = x_1 + x_2 + x_3 + \cdots + x_n$$

Later this will become so commonplace that we'll adopt an even shorter shorthand.

1.2.1 Vectors and Matrices

Thinking initially of vectors as mere collections of numbers (a viewpoint which we will rethink in §3), indices give us a way to talk about them. x_n could represent a single row or column of n numbers.

Likewise we can use two indices to label the numbers in a grid or matrix (plural: matrices). Given:

$$M = \begin{bmatrix} 5 & 2 & 7 \\ 0 & 1 & 0 \\ 4 & 6 & 8 \end{bmatrix}$$

We can refer to the elements of M as M_{ij}, with i giving the row and j the column. So:

$$M_{32} = 6$$

This unfortunately looks a lot like the number 32. When it's clear we aren't talking about raising numbers to powers, we use a combination of subscript and superscript indices, with superscripts meaning rows and subscripts meaning columns:

$$M^3_2 = 6$$

1.2.2 Matrix multiplication

The simplest introduction to the purpose of a matrix is to consider a transformation that operates on the Euclidean plane, mapping any point given by coordinates (p_1, p_2) to new positions given by (q_1, q_2). We will restrict ourselves to transformations that can be expressed as:

$$\begin{aligned} q_1 &= M_{11}p_1 + M_{12}p_2 \\ q_2 &= M_{21}p_1 + M_{22}p_2 \end{aligned} \tag{1.1}$$

That is, we only choose four numbers $M_{ij} = (M_{11}, M_{12}, M_{21}, M_{22})$ to completely control the transformation. In matrix notation the above is:

$$\begin{aligned} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} &= \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \\ &= M \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \end{aligned} \tag{1.2}$$

Or as a summation, where the i and j can take on the values 1 or 2:

$$q_i = \sum_j M_{ij} p_j \tag{1.3}$$

What if we then apply to (q_1, q_2) a second such transformation described by $N_{11}, N_{12}, N_{21}, N_{22}$ to get (r_1, r_2)?

$$\begin{aligned}\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} &= \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \\ &= \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \left(\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \right)\end{aligned} \tag{1.4}$$

By returning to the actual formulae and doing the substitution we can arrive at the answer, via a mess that ends up very simple. Doing the tedious work for r_1 alone:

$$\begin{aligned} r_1 &= N_{11}(M_{11}p_1 + M_{12}p_2) + N_{12}(M_{21}p_1 + M_{22}p_2) \\ &= N_{11}M_{11}p_1 + N_{11}M_{12}p_2 + N_{12}M_{21}p_1 + N_{12}M_{22}p_2 \\ &= (N_{11}M_{11} + N_{12}M_{21})p_1 + (N_{11}M_{12} + N_{12}M_{22})p_2 \end{aligned} \tag{1.5}$$

Doing the same for r_2, it turns out that applying these two transformations is like applying a single transformation given by another matrix O that we can prepare directly from M and N:

$$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} N_{11}M_{11} + N_{12}M_{21} & N_{11}M_{12} + N_{12}M_{22} \\ N_{21}M_{11} + N_{22}M_{21} & N_{21}M_{12} + N_{21}M_{22} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$$

Which means that instead of bracketing how we did in (1.4), we can "multiply" the two matrices first:

$$\begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \begin{bmatrix} N_{11}M_{11} + N_{12}M_{21} & N_{11}M_{12} + N_{12}M_{22} \\ N_{21}M_{11} + N_{22}M_{21} & N_{21}M_{12} + N_{21}M_{22} \end{bmatrix} \tag{1.6}$$

Generalising, if we're multiplying two matrices M and N to get O:

$$O = MN$$

in summation notation we can define the cell at row i and column j of O as follows:

$$O_{ij} = \sum_k M_{ik}N_{kj}$$

This formula describes all the above examples of matrix multiplication. If j is only allowed to take on the value 1 then N and O become column matrices with no need of a second index, and we've recreated (1.3) albeit with some light renaming.

Note that in general $MN \neq NM$, that is, multiplication is not necessarily commutative, because referring to the result of (1.6) we can see there is no way to rearrange the terms to make the two results equal. However they can be made to match up if the matrices are symmetric, meaning that diagonally opposite elements are equal ($M_{ij} = M_{ji}$), or equivalently the matrix is equal to its transpose ($M = N^T$), so in that case there is commutativity.

But also matrix multiplication turns out to be merely one combination of some more basic concepts we'll return to in §3.4.4.

1.2.3 Kronecker Delta

One of the most important examples of a matrix is the *identity* that makes no difference when it appears in a multiplication. Referring to (1.1), we want $q_i = p_i$ and therefore the main diagonal elements M_{ii} need to be 1, and all others are 0. This generalises to any size of square matrix, but is awkward to represent in a stretchy way:

$$\hat{I} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

So instead we define the Kronecker delta, which has two indices representing row and column:

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases}$$

Although in this case it's not important which index is the row and which the column, due to the symmetry of the identity matrix.

1.2.4 Manipulating Summations

As a summation is just a template for generating terms that added, the algebraic manipulations we casually use all the time for addition and multiplication are automatically available. Multiplication is distributive across addition, so:

$$\sum_n x f(n) = x \sum_n f(n)$$

where x is a constant. Also, think about "multiplying out" two multi-term expressions:

$$(a_1 + a_2 + a_3)(b_1 + b_2) = a_1b_1 + a_1b_2 + a_2b_1 + a_2b_2 + a_3b_1 + a_3b_2$$

So in a summation we can name two indices and this produces a term for every combination of their allowed values:

$$\left(\sum_i a_i\right)\left(\sum_j b_j\right) = \sum_{ij} a_i b_j$$

This works in both directions. If we initially have a summation expression:

$$\sum_{kl} a_k b_l$$

and then we discover that a_k is itself actually the result of another summation:

$$a_k = \sum_j M_{kj}$$

(and here, note that k is acting like an integer parameter of a function), we can obviously perform a substitution:

$$\sum_{kl}\left(\sum_j M_{kj}\right) b_l$$

Unfolding that, supposing all three indices may take the values $\{1, 2\}$:

$$\begin{aligned}&(M_{11}b_1 + M_{12}b_1)+\\ &(M_{21}b_1 + M_{22}b_1)+\\ &(M_{11}b_2 + M_{12}b_2)+\\ &(M_{21}b_2 + M_{22}b_2)\end{aligned} \tag{1.7}$$

Each of the four parenthesised pieces results from one expansion of the inner sum over j, but the parentheses can melt away leaving eight terms added together, which can be written as:

$$\sum_{jkl} M_{kj} b_l$$

But equally we could begin with the above and extract the concept of summing a slice of M by inventing:

$$a_k = \sum_j M_{kj}$$

Which has the effect of removing the j summation index, as it's hidden within the computation of a_k:

$$\sum_{kl} a_k b_l$$

These may appear counter-intuitive at first glance but when you remember what summation expands into, they are quite obvious.

1.3 Expectation Value

This unfortunate statistical term is used everywhere; unfortunate because it describes a value that we do not necessarily expect to ever measure, and even more unfortunate that it is often garbled into "the expected value", which may be entirely untrue. It is the expected *mean* of a set of repeated measurement values.

For a set of discrete values taken by some integer variable n, the values may be $2, 3, 3, 3, 4, 4, 5$, which sums to 24, and there are 7 values, so the mean value is $3.42857...$ which is not an integer so clearly cannot be an expected value.

Looking at the list of values, we can tabulate them by giving the observed ("frequentist") probability P of each value (number of times it occurs divided by the size of the set of values):

n	P_n
2	$1/7$
3	$3/7$
4	$2/7$
5	$1/7$

P_n is zero for all n except the above exceptions, where it is between zero and 1, and of course all values of P_n add up to 1 because we fixed them to do that when we divided them all by 7. P_n is literally "what fraction of the 7 values is contributed by n".

Therefore by computing the weighted sum:

$$\langle n \rangle = \sum_n n\, P_n$$

we recover the mean value $\langle n \rangle$. The point here is that, inside a sum at least, it makes sense to multiply a value by the probability of obtaining that value.

In the continuous case, the probability density function $\rho(x)$ does not give us the probability of x, a meaningless concept for a continuous variable (any specific value is infinitesimally unlikely), but it can be integrated over some region to get the probability of the value appearing in that region.

The integral over all values of x:

$$\langle x \rangle = \int_{-\infty}^{+\infty} x\, \rho(x)\, dx$$

is the continuous equivalent of (1) and gives the mean value of a large set of measurements of x. If we think of all the values of x as a cloud of matter that is more or less densely concentrated here or there, $\langle x \rangle$ is like its centre of mass.

But $\rho(x)$ may be symmetrical around the origin and vanish at the origin, e.g. two peaks on either side, making $\langle x \rangle = 0$ despite x never taking the value 0; so if we are required to call it "the expectation value", we must always remember that it may be a value that never occurs.

1.4 Fourier Transform

Given a real-valued function $f(x)$, and supposing it is periodic, e.g. it describes the sound of a bell ringing, you might ask what frequencies appear in the sound. In fact your ear-brain system is an adaptation for answering that very question, and if you listen carefully you can often discern several different notes within the sound of a bell.

What we're really asking is how "loud" the signal is at each frequency. We can detect this for a given frequency ν by multiplying the function by $e^{-i2\pi\nu x}$, in which:

- the minus sign is purely a convention (and not a universal one)
- i is the magic ingredient that makes it go round and round
- 2π converts to radians
- ν is the frequency
- x is the parameter to the function

So if ν is 1, the complex value performs a whole rotation as x goes from 0 to 1, and again from 1 to 2, etc.

By itself this factor is a unit complex number, i.e. of "length" 1, but by multiplying it by the function we adjust its length so it oscillates "in and out" as it rotates, exactly like our signal:

$$f(x)e^{-i2\pi\nu x}$$

If the oscillations of f don't coincide with the frequency ν, the above expression will, averaged over all values of x, be about zero, there being no particular reason for the complex value to be biased in any direction. That is:

$$\int_{-\infty}^{\infty} f(x)e^{-i2\pi\nu x}dx \approx 0$$

But if the oscillations do coincide, then there will be a bias; each time the oscillation of $f(x)$ reaches a maximum it will be on the same side of the circle traced by $e^{-i2\pi\nu x}$.

(A minor subtlety is that whenever $f(x)$ is at a negative minimum, $e^{-i2\pi\nu x}$ will be on the other side of the circle; however, multiplying it by the negative value of $f(x)$ will flip it round by 180 degrees, so both positive peaks and negative troughs will both contribute to the same biased direction.)

So we can define a complex-valued function of frequency:

$$\hat{f}(\nu) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi\nu x}dx \tag{1.8}$$

and this will be about zero for frequencies that don't appear in the function, and non-zero for frequencies that do appear. These values are *complex* amplitudes; they tell us how loud the signal is at that frequency, but also their phase tells us how the signal is offset at that frequency.

As a shorthand we can write it as a fancy $\mathcal{F}$:

$$\hat{g} = \mathcal{F}g$$

We can do the opposite transformation:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\nu)e^{i2\pi\nu x}d\nu \tag{1.9}$$

Shorthand:

$$g = \mathcal{F}^{-1}\hat{g}$$

This pretty much literally says that you can make any function by adding together an infinite collection of oscillations at every possible frequency. You just need a (complex) function $\hat{f}(\nu)$ that tells you how "loud" each frequency needs to be.

1.4.1 Negative Frequencies

By the way, note how when we do the integral over $\hat{f}$ in the inverse transform, we include negative values. What on earth is a negative frequency?! It's not that weird, really. It just makes the complex factor rotate the other way. This underscores the fact that the minus sign is just a convention. The integrals cover both "directions".

There's a special relationship between the positive and negative sides of the frequency spectrum. According to equation (1.8), if you want the amplitude for frequency $-\nu$, it must be:

$$\hat{f}(-\nu) = \int f(x) e^{i2\pi\nu x} dx$$

Which means $\hat{f}(-\nu)$ is the complex conjugate of $\hat{f}(\nu)$.

$$\hat{f}(-\nu) = \left[\hat{f}(\nu)\right]^*$$

Therefore having done the hard work of computing one side, it is very easy to get the other side - it contains no different information.

Also, note that as we're integrating over all x from $-\infty$ to ∞, we can negate x through the integral without changing the result. So this produces the exact same frequency amplitudes as equation (1.8):

$$\hat{f}(\nu) = \int f(-x) e^{i2\pi\nu x} dx$$

Suppose f happens to be an even function:

$$f(-x) = f(x)$$

Then we can switch freely:

$$\hat{f}(\nu) = \int f(x) e^{i2\pi\nu x} dx$$

If we mirror the frequency:

$$\hat{f}(-\nu) = \int f(x) e^{-i2\pi\nu x} dx$$

But we've arrived back at equation (1.8), meaning it must be perfectly symmetrical around $\nu = 0$ when applied to an even function. This means it must also be real at all frequencies - how else could all this be true?

$$\hat{f}(-\nu) = \hat{f}(\nu) = \left[\hat{f}(\nu)\right]^*$$

Now suppose f is odd:

$$f(-x) = -f(x)$$

By a similar argument, when we substitute:

$$\hat{f}(\nu) = -\int f(x)e^{i2\pi\nu x}dx$$

And mirror:

$$\hat{f}(-\nu) = -\int f(x)e^{-i2\pi\nu x}dx = -\hat{f}(\nu)$$

If taking the complex conjugate is the same as negating, we must be talking about a purely imaginary number. So the transform of an odd function is imaginary and odd.

1.4.2 Spikes

What happens if we take the Fourier transform of a pure $\sin$ wave? Only a single frequency is present. To describe this situation, the mathematical tool we need is called the Dirac delta, $\delta(x)$, and is often referred to as a function, or a "function" with scare-quotes. It has a number of strange properties if regarded as a function, so it's simpler to think of it as only ever appearing as a factor inside an integral. But in simple terms, it is zero except at $x = 0$, where it is infinite. We can use an expression like $\delta(x - \alpha)$ to move the spike from zero to the location α of our choice.[1]

Why does it have to be infinite at the spike? We recover the function from its transform with the inverse transform, which is an integral:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\nu)e^{i2\pi\nu x}d\nu$$

Substituting $\delta(\nu - \alpha)$ as the transform from which we're recovering the function, i.e. a spike at α:

[1] It's like the real number equivalent of the Kronecker delta, though we write that slightly differently. You can think of the Kronecker δ_{nm} (§1.2.3) as conceptually similar to the Dirac $\delta(n-m)$: if $n \neq m$, the result is 0.

$$f(x) = \int_{-\infty}^{\infty} \delta(\nu - \alpha) e^{i2\pi\nu x} d\nu$$

Imagine the value of ν sweeping through the range of values from $-\infty$ to $+\infty$, everywhere contributing nothing except at the instant it passes through $\nu = \alpha$. That contributes $e^{i2\pi\alpha x}$. To accomplish that, the other factors must have the product 1:

$$\delta(\nu - \alpha) d\nu = 1$$

and so:

$$\delta(\nu - \alpha) = \frac{1}{d\nu}$$

So the spike at α from $\delta(\nu - \alpha)$ must be infinite so as to counteract the infinitesimal smallness of $d\nu$. In other words, we have to think about any transform as an amplitude *density* function.

1.4.3 The Gaussian

An infinitesimally narrow spike in the frequency spectrum represents the single frequency present in a pure wave that goes on forever. And a pure wave in the frequency spectrum represent the infinite set of frequencies that must be summed to get an infinitesimally narrow spike. Each is the Fourier transform of the other. They are the two extremes of:

- being localised in position but spread out in frequency - being spread out in position but localised in frequency

Between these two there is a middle ground, a shape that is its own Fourier transform. The best known example is the Gaussian, of the form:

$$g(x) = Ae^{Bx^2}$$

Where A and B are constants. There are several ways of concluding that its Fourier transform is:

$$\hat{g}(\nu) = A\sqrt{\pi/B} e^{-\frac{\pi^2}{B}\nu^2}$$

This is evidently of the same form. Aside from being a function of frequency ν instead of position x, the constant A has become $A\sqrt{\pi/B}$, and B has become $-\frac{\pi^2}{B}$, and these are just different constants.

1.5 Potentials

A force field is a vector-valued function of space, i.e. at each point in space we imagine there is a vector giving the strength and direction of the force that would felt at that point.

The force fields we observe in nature have an interesting property: it is always possible to replace the force field with a scalar-valued function of space, i.e. at each point in space there is merely an ordinary number, not a vector. We can then take the vector gradient ∇ of this scalar field and we recover the force field.

By analogy, picture a hilly landscape. The height H above sea level is the scalar field value, so the landscape is fully described by the scalar field $H(x, y)$. From this we can derive ∇H, a two-dimensional vector field (picture it as an arrow that never points up or down, always parallel to the horizon). As we travel around we sometimes face steep slopes, where ∇H points in the steepest direction, or stationary points such as hilltops or valley basins where ∇H is the zero vector (to distinguish between peaks and valleys, we'd need to take the second derivative, $\nabla^2 H$).

If we wander on some pathway through this landscape and return back to where we started, our height will be the same as it was when we started (assuming the landscape hasn't changed shape). This is true regardless of the path we take, as the height is a fact about the start/end point of the path. This is so obvious as to seem hardly worth stating.

And yet if we only had some vector field, and wondered if the path integral of any closed loop through that field was always zero, how would we know? Some paths might go mainly through regions with vectors all pointing in one direction, and so not sum to zero. Not all vector fields have this self-balancing property.

Those that do are known as conservative fields, and these are fields which can be reduced to a scalar field from which the vectors can be recovered by applying ∇, and these are all the force fields we encounter in nature.

When we describe a force field by a scalar field, we call that field a *potential*. It has units of energy. As a particle moves through a potential, it experiences a potential difference between two points. If this difference is negative, i.e. the potential energy drops between the two points, the particle gains kinetic energy (speeds up). This is exactly like a ball rolling down a slope; the potential energy is exactly equivalent to the height of the landscape.

If the potential does not vary, the gradient is zero. This is true regardless of the potential's constant value, which is like a constant of integration, i.e. a global increase in potential is physically meaningless.

An important example is a force field conforming to the inverse square law, so

the force is proportional to r^{-2} where r is the distance from the origin of the force. The potential must therefore be proportional to r^{-1}, so that it has the required gradient (differentiation subtracts 1 from the power of a polynomial).

Chapter 2

Unit Circle

2.1 Introduction

The unit circle (Figure 2.1) is a treasure trove of strange connections with other areas of mathematics, many of which are constantly useful in physics.

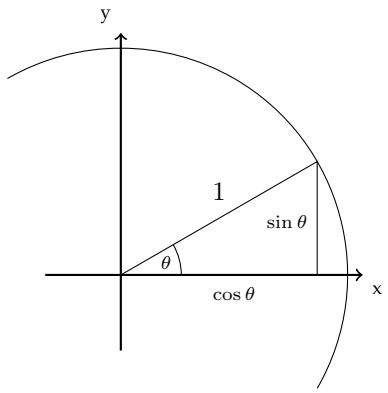


Figure 2.1: The unit circle

The points that make up the circumference (edge) of a unit circle are all the points that are 1 unit of distance from the centre. The same definition is used for a sphere in any number of spatial dimensions.

For historical reasons[1] when measuring distances travelled around the circumference of a unit circle we define the number π to be the length of half a circumference, so length of the circumference of a unit circle is 2π. We call this

[1]The ratio between the diameter, d of a circle and the circumference is π. But it has turned out over succeeding millennia that the radius $r = d/2$ is far more commonly encountered in calculations, and so we are doomed to say 2π almost everywhere.

distance travelled around a part of the unit circle's circumference the *angle*. By convention we always begin measuring the angle from the x-axis and moving in the counter-clockwise direction. In the figure the angle θ happens to be $\pi/6$, the same as $30°$.

2.2 Sine and Cosine

Given an angle, it is surely possible to compute the (x, y) coordinates of the corresponding point on the circumference. There must exist a pair of functions that give these coordinates, traditionally named as follows:

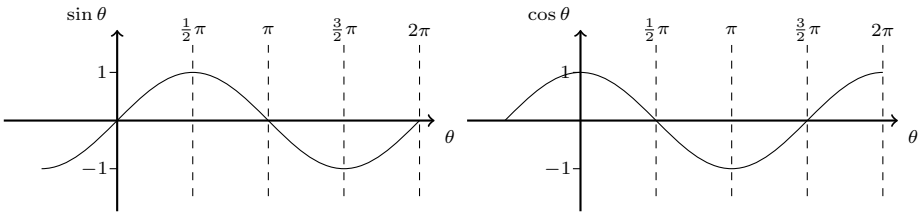
- $\sin\theta$ ("sine") gives the vertical or y coordinate, and
- $\cos\theta$ ("cosine") gives the horizontal or x coordinate.

Without knowing anything else about these functions, we can already see that as an object progresses around the circle, its coordinates must visit these milestones:

- We begin at $\theta = 0$, the right-most point of the circle, where $\sin\theta = 0$ and $\cos\theta = 1$.
- At $\theta = \pi/2$ (90°), the top of the circle, the situation has reversed: $\sin\theta = 1$ and $\cos\theta = 0$.
- At $\theta = \pi$ (180°), the left-most point of the circle, $\sin\theta = 0$ and $\cos\theta = -1$.
- At $\theta = 3\pi/2$ (270°), the bottom of the circle, $\sin\theta = -1$ and $\cos\theta = 0$.

We can plot $\sin\theta$ by itself (Figure 2.2a) between the values 0 and 2π, and likewise $\cos\theta$ (Figure 2.2b).

Figure 2.2: Sine and Cosine



(a) The $\sin$ function (b) The $\cos$ function

We can see that $\cos$ is just $\sin$ advanced by a quarter of a cycle, or:

$$\cos\theta = \sin(\theta + \pi/2)$$

If we advanced $\sin$ by π, half a cycle, the peaks and valleys would change places, while the uphill slopes that cut through the horizontal axis would become downhill slopes and vice versa. In other words, it would look exactly like we'd flipped the $\sin$ function upside down:

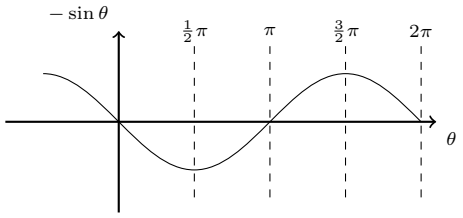


Figure 2.3: $\sin\theta$ upside down

So:

$$-\sin\theta = \cos(\theta + \pi/2) = \sin(\theta + \pi)$$

Incidentally, the word *cycle* is from the Greek word for circle. The word *phase* is also used a lot in this context, probably deriving from the phases of the moon. When one periodic function is equal to another shifted by some amount, we say there is only a phase difference between them.

2.3 Differential Calculus on Sine and Cosine

Think of an object moving along the unit circle's edge at a constant rate. It has an instantaneous velocity vector pointing along the tangent, but this is always at a right angle to the line segment reaching from the centre to the object (the radius).

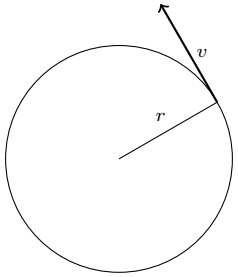


Figure 2.4: The velocity vector v of an orbiting object

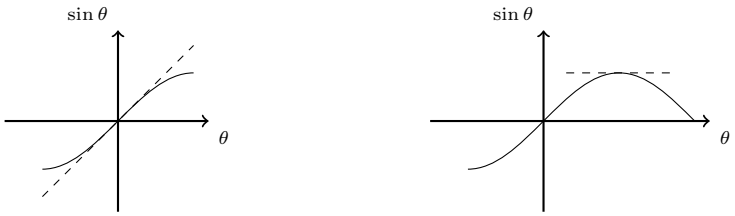
The velocity vector is therefore fixed to be a quarter cycle ahead of the radius, which is precisely the relationship between $\cos\theta$ and $\sin\theta$. The components of v are the rates at which the corresponding components of r change as the angle

advances. It follows that the rate at which $\sin\theta$ is increasing is given by $\cos\theta$, or:

$$\frac{d}{d\theta}\sin\theta = \cos\theta$$

This is confirmed by examining the shape of the functions separately. Nearby to $\theta = 0$, $\sin\theta$ is closely approximated by θ itself, and looks (Figure 2.5a) much like a straight line of gradient 1, which happens to be the value of $\cos 0$.

Figure 2.5: Gradients of Sine



(a) $\sin\theta \approx \theta$ (b) Gradient 0

When $\sin\theta$ peaks at $\theta = \pi/2$, its gradient is 0, and again this is in agreement with the value of $\cos(\pi/2)$. This pattern continues at every point in the cycle.

So in any situation where we need the derivative of $\sin$, we just replace it with $\cos$. By the same reasoning (and because we know the same relationship exists between $-\sin\theta$ and $\cos\theta$) we can assume that:

$$\frac{d}{d\theta}\cos\theta = -\sin\theta$$

Differentiating two more times will get us back to $\sin\theta$ because each differentiation advances the function by a quarter cycle. Naturally integration performs the same trick in the opposite direction.

2.4 Computing Sine and Cosine

Now we can take the derivative of $\sin$, and we know its precise value at 0 (and those of its derivatives), we can use the Maclaurin method to find how to compute it using only addition and multiplication, if only we have the patience to perform an infinite sequence of those operations. We begin by assuming that a function $f(x)$ can be expressed as the sum of an infinite series of polynomial terms, that is, some constant a_n multiplied by the variable x raised to an integer power n, beginning at 0:

$$f(x) = a_0x^0 + a_1x^1 + a_2x^2 + a_3x^3 + \ldots$$

Anything raised to the power of 0 is 1, and raising something to the power of 1 makes no difference, so at the expense of consistency, and meanwhile also naming the function we want to compute, we just write:

$$\sin x = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots$$

So we only need to determine the constant factors a_n to entirely characterise the function. Setting $x = 0$, any term that is multiplied by x must be zero (they *vanish* entirely from the sum), which only leaves a_0. But $\sin 0 = 0$, so it follows that $a_0 = 0$. Taking the derivative of both sides:

$$\cos x = a_1 + 2a_2 x + 3a_3 x^2 + 4a_4 x^3 + \dots$$

Each polynomial's power is reduced by 1, and the original power "moves down" to become an extra factor. Now the exact same vanishing argument applies to a_1 except that here it is solely responsible for the value of $\cos 0 = 1$, and so it must be that $a_1 = 1$. We differentiate a second time:

$$-\sin x = 2a_2 + (3 \cdot 2)a_3 x + (4 \cdot 3)a_4 x^2 + \dots$$

So $a_2 = 0$. Differentiate a third time:

$$-\cos x = (3 \cdot 2)a_3 + (4 \cdot 3 \cdot 2)a_4 x + \dots$$

So $a_3 = -\frac{1}{3!}$. When n is even the term is zero, and when n is odd the term x^n is divided by the accumulated value $n!$. We continue in this fashion to find that:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

By the same process we find that for cos only the even-n terms appear in the result:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

In each case there is a four stage cycle, with alternating absence of terms and then similarly alternating $+/-$ signs. This is curiously reminiscent of the cyclic behaviour of imaginary unit number i, defined such that $i^2 = -1$, as we raise it to integer powers starting at zero:

- $i^0 = 1$
- $i^1 = i$

- $i^2 = -1$
- $i^3 = i(i^2) = -i$
- $i^4 = (i^2)(i^2) = (-1)(-1) = 1$
- and so on: i, -1, $-i$, 1, i, -1 ...

2.5 Euler's Formula

If we get carried away by the power of this method we risk drifting from the topic of the unit circle, but it turns out that there is no escape.

The exponential function e^x raises the constant e to a variable power, and the value e is chosen such that the derivative is e^x, so the function is its own derivative. The method of evaluating at zero and taking repeated derivatives is therefore particularly simple to perform and yields:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$$

This is obviously different from the previous two examples because no terms vanish and all terms are positive. But what happens if we replace x with xi?

$$e^{ix} = 1 + ix + \frac{i^2x^2}{2!} + \frac{i^3x^3}{3!} + \frac{i^4x^4}{4!} + \frac{i^5x^5}{5!} + \dots$$

Simplifying the various powers of i according to its definition:

$$e^{ix} = 1 + ix - \frac{x^2}{2!} - i\frac{x^3}{3!} + \frac{x^4}{4!} + i\frac{x^5}{5!} + \dots$$

It appears we can write it as the sum of two series, one containing only real terms and the other only imaginary:

$$e^{ix} = (1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots) + i(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots)$$

But the real component is evidently $\cos x$ while the imaginary one is $\sin x$!

$$e^{ix} = \cos x + i \sin x$$

This means that x is the angle, so we will resuming calling it θ. We can interpret $e^{i\theta}$ as a unit circle in the complex plane. It is all the complex numbers of modulus 1.

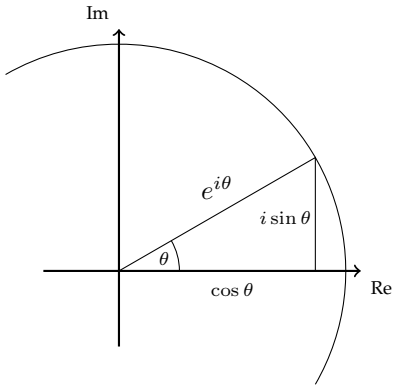


Figure 2.6: The unit circle in the complex plane

We can relabel the axes on our diagram of the unit circle, to show the circle in the complex plane (Figure 2.6).

We can scale the circle to any radius r simply by multiplying by that radius. This means we can represent any complex number either as a sum $x + yi$, or as an exponential $re^{i\theta}$.

Physics is thick with examples of oscillation. To describe the state of a particle undergoing simple harmonic motion along a line, we need to know the position and the momentum. But a complex number, being described by two real numbers, can encapsulate both these quantities, and the real and imaginary parts have the right phase relationship.

2.6 Pythagoras in the Circle

There is a right-triangle in the diagram, so by Pythagoras:

$$x^2 + y^2 = 1$$

it must be that:

$$(\sin \theta)^2 + (\cos \theta)^2 = 1$$

Also by Pythagoras, given one coordinate of a point on the circle, we can compute the other:

$$y = \sqrt{1 - x^2} \quad x = \sqrt{1 - y^2}$$

Chapter 3

From Vectors to Tensors

3.1 What we need to unlearn

We are first introduced to vectors in two different yet closely related and simplified forms. We're now going to rethink them as an abstraction, which will require us to be careful not to depend on any intuitions derived from our earlier encounters.

That's not to say that we will be abandoning the schoolhouse version of vectors; rather, we will be properly placing them in the context of a more general framework. Also they will very often help ground us, as long as we recognise their limitations.

3.1.1 Arrows with direction and length

The first way to think of vectors is by visualising them as arrows that have a direction and a length. Two vectors $\mathbf{a}$ and $\mathbf{b}$ can be added (Figure 3.1) by laying them head to tail, so the sum $\mathbf{c}$ is the vector starting at the tail of $\mathbf{a}$ and ending at the head of $\mathbf{b}$.

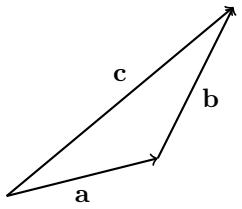


Figure 3.1: Adding arrows.

Scaling a vector (multiplying it by a number) just alters its length without

changing its direction, e.g. multiply by 0.5 to shrink the vector to half its prior length.

We also learn about the dot product, a scalar-valued operator between two vectors, $\mathbf{p} \cdot \mathbf{q}$. If the two vectors $\mathbf{p}$ and $\mathbf{q}$ are separated by angle θ, and we know the magnitude (length) of each vector, e.g. $\|\mathbf{p}\|$, then:

$$\mathbf{p} \cdot \mathbf{q} = \|\mathbf{p}\|\|\mathbf{q}\| \cos\theta$$

When we get onto the abstract definition of a vector it may seem like the geometric viewpoint has been relegated to a special case, less fundamental. But it is often useful to keep it in your mind as a way to visualise vectors of any kind, because however abstractly they are defined, they will always be closely analogous to the familiar arrows.

3.1.2 Columns of numbers

The second concrete way to think of vectors is as columns of ordinary numbers, and the number of *dimensions* of the space tells us how many numbers a column vector has to contain. In this form, to add two vectors we just deal with the rows separately: add the numbers in row 1, and then the numbers in row 2 and so on for however many rows there are in a column vector, and thus obtain the sum as a column:

$$\begin{bmatrix} 2 \\ 0.5 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2.5 \end{bmatrix}$$

More succinctly we can use index notation a_n to mean the value in the nth row of the column associated with vector $\mathbf{a}$, so to add two vectors we just do this:

$$a_n + b_n = c_n$$

Scaling a vector just involves multiplying all the rows by the same number:

$$d_n = x a_n$$

The dot product is extremely simple in this representation: like with addition, you treat each row separately, multiplying the numbers in row 1 and so on, but then you just sum all the products to get the numeric value:

$$\sum_n a_n b_n$$

3.1.3 Coordinates

These two perspectives are united by introducing a coordinate grid (Figure 3.2).

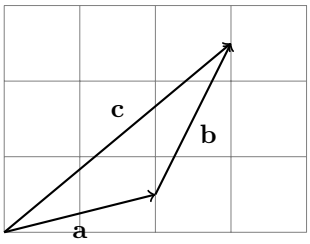


Figure 3.2: Coordinate grid.

Much of this subject is concerned with ensuring that our choice of coordinate grid doesn't get confused with the physical facts. We're trying to get answers about nature, and those answers better not change just because we used a different coordinate grid. One of the most important ideas in physics is that vectors are primarily geometric objects. They can be described with numeric coordinates, but there is no preferred coordinate basis. A vector has an independent existence, because it describes something in the physical world.

But however abstract things get, it can often be helpful to remember that you can visualise vectors as arrows and think of the basic operations on them geometrically, and equally it can be helpful to remember that we will always have a way of representing vectors as columns of numbers (indeed, columns of numbers *are* vectors.)

The primary intuition we've been implicitly relying on so far is *orthonormality*. With arrow vectors we can simply see when they are orthogonal, or to be more precise we can measure the angle between vectors, and we can measure their lengths, and we can choose a unit length, and so on. We can simply draw a unit vector, and then draw another unit vector that is orthogonal to it.

Likewise from the column vectors we have no difficult choosing a set of orthonormal vectors, the *standard basis*. They are *one-hot*, all values zero except for a single 1. In an n-dimensional space there can only be n such distinct vectors.

None of these intuitive leaps will be available with abstract vector spaces, and orthonormality cannot be used as an elemental building block. We will build a quite rich set of more fundamental concepts before we invent orthonormality.

By the way, when in physics we speak of a *vector field*, that is, a vector at each point in space, such as wind speed and direction, or the electric field, we visualise arrows spread out over space. But the value of the field in two different places may be the same.

This is obvious (and less confusing) in the case of a scalar field, such as temperature. At two different locations in a room, the temperature may be the same. It's a numerical value that varies from place to place, and the same number may appear in two places.

But exactly the same is true for a vector field. If the wind is some particular speed and direction at two different places on the map, we say the vectors are equal: they are the *same vector*. From the point of view of considering their equality, it is irrelevant that they are associated with different locations in physical space. In vector space, there is one vector with that direction and length.

On to the abstract stuff.

3.2 Vectors as elements of a vector space

A vector space is a set of objects, called vectors, about which we assume nothing except that we can perform certain operations on them.

3.2.1 They can be added

There is an operator $+$ that takes two objects from the set and returns another from the same set (we say it's a *closed* operator).

This operator is commutative:

$$\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$$

and associative:

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

There is a special object called 0 (the *zero vector*), which makes no difference when added to any object from the set:

$$\mathbf{v} + 0 = \mathbf{v}$$

Also every object has an opposite, known as its additive inverse, so they pair up. The inverse of $\mathbf{v}$ is written as $-\mathbf{v}$, and:

$$\mathbf{v} + (-\mathbf{v}) = 0$$

The above can written as $\mathbf{v} - \mathbf{v}$. Evidently 0 is its own inverse.

Referring back to schoolhouse vectors, we can see how the arrows and the columns have an addition operation that satisfies all these requirements.

3.2.2 They can be scaled

There is an associated set of objects called scalars, typically restricted to real or complex numbers. Our objects can be multiplied by a scalar to get another object. Scaling them by 1 makes no difference. Scaling them by -1 discovers the additive inverse.

Given two scalars a and b, we can compute $c = ab$ and then scale an object $\mathbf{v}$ by it, or we can separately scale the object first by a and then by b, and the result is the same:

$$(ab)\mathbf{v} = a(b\mathbf{v})$$

Scaling is distributive over addition of objects:

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

And also over addition of scalars:

$$(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$$

Again, arrows and columns have no problem meeting these requirements.

3.2.3 Other Examples of Vector Spaces

Any set of objects for which we can define these operations is a vector space, not just arrows and columns. The set of ordered tuples of real numbers $\mathbb{R}^n$ is just the column vectors with n rows each. Also there is no reason why n shouldn't be 1, which means that the plain old set of real numbers $\mathbb{R}$ is also vector space. Think of the real number line as My First Vector Space™.

Also the complex numbers $\mathbb{C}$, and tuples of them $\mathbb{C}^n$, work just as well. The example of $\mathbb{C}$ as a vector space is particularly interesting because of its close similarly to $\mathbb{R}^2$. The major difference is that it has a definition of multiplication as a closed operation over its vectors (such that the product of two vectors is a vector), which is absolutely not a general feature of vector spaces.[1]

In quantum mechanics we will contend with infinite-dimensional complex vector spaces.

3.2.4 Fields

The kind of set that can serve as a scalar is called by mathematicians a *field* (an unfortunate collision of terminology given the very different meaning in

[1] Although it is also defined (very differently) in $\mathbb{R}^3$ as the cross product, $\times$.

physics), which is a set of objects on which we have defined addition, subtraction, multiplication and division, so real or complex numbers usually serve this purpose (and always do in physics), but vectors in general cannot serve as a field of scalars for other vector spaces, because the definition of a vector space says nothing about there being a natural way to multiply or divide pairs of vectors to obtain other vectors.

In the same way, you can't have a vector space of $\mathbb{R}$ over the field of $\mathbb{C}$, because although we can use regular multiplication to "scale" a vector from $\mathbb{R}$ by a scalar from $\mathbb{C}$, the result is likely to be a member of $\mathbb{C}$ but not of $\mathbb{R}$, and thus not a vector from the same space.

Unless we say otherwise, we'll assume the field is $\mathbb{R}$.

3.2.5 Finding a Basis

If we select two vectors $\mathbf{a}$ and $\mathbf{b}$ from the space, we may find that they only differ by a scalar ratio x:

$$\mathbf{a} = x\mathbf{b}$$

If there is an x that can scale $\mathbf{b}$ into $\mathbf{a}$ then those two vector are *colinear*[2] (we are careful not to say they point in the same direction because if x is negative then they point in exactly opposite directions, but are still colinear.)

But if there is no such x then they are not colinear. This gives them an interesting superpower:

$$\mathbf{r} = x\mathbf{a} + y\mathbf{b}$$

By varying the scalar coefficients x and y we can construct any vector $\mathbf{r}$ in a two-dimensional *subspace* of the vector space.

We can generalise on this idea a bit by rearranging the equation (and supposing that x becomes negative). If for two vectors $\mathbf{a}$ and $\mathbf{b}$ we can find a scalar x so that:

$$\mathbf{a} + x\mathbf{b} = 0$$

then they are colinear. Suppose the two vectors point in the same direction but $\mathbf{a}$ is twice the length of $\mathbf{b}$. Then we can set $x = -2$ and the sum will cancel out. This is only possible because they are colinear. If it's not possible, we've found a pair of *linearly independent* vectors. Now we can look for a third:

[2]This literally means "on the same line". Note the interesting use of geometrical language, even though we're not supposed to be thinking about arrows in this abstract discussion

$$\mathbf{a} + x\mathbf{b} + y\mathbf{c} \neq 0$$

Supposing we find such a vector $\mathbf{c}$ for which there is no scalar y that satisfies the above equation, then we have found three linearly independent vectors. Or to put it another way, it is not possible to make $\mathbf{c}$ by any weighted sum of $\mathbf{a}$ and $\mathbf{b}$:

$$\mathbf{c} \neq x\mathbf{a} + y\mathbf{b}$$

Now we can construct any vector in three dimensions:

$$\mathbf{r} = x\mathbf{a} + y\mathbf{b} + z\mathbf{c}$$

Eventually we may find (assuming the space is finite dimensional) that it is not possible to extend our linearly independent set $\mathbf{a}, \mathbf{b}, \mathbf{c}, ...$ any further. The size of this set tells us how many dimensions the space has, and these vectors are said to *span* the space.

The vitally important thing to realise about this is that at no point have we said that these vectors are orthogonal. We haven't even defined what that means yet. We've only defined the property of linear independence. Nevertheless we have arrived at the idea of a coordinate grid; it's just that our grid may be awkwardly slanted, made of identical parallelogram tiles rather than identical square tiles.

So we don't have to keep choosing letters, we will label each dimension with a number. The set of linearly independent vectors that we can use to construct any other vector in the space is called a *basis*. The basis vectors are traditionally written as $\mathbf{e}_n$, where n is often 1-based (although in Relativity it may be 0-based; this is purely a notational convention and makes no arithmetic difference). The scalar coefficients, which we will call coordinates, that construct a given vector $\mathbf{r}$ can also be numbered, conventionally with superscript r^n:

$$\begin{aligned} \mathbf{r} &= r^1\mathbf{e}_1 + r^2\mathbf{e}_2 + ... + r^n\mathbf{e}_n \\ &= \sum_n r^n\mathbf{e}_n \end{aligned} \tag{3.1}$$

The use of a superscript index is obviously asking for trouble given that it looks like we're raising r to a power[3], but this notation is universal in physics so we may as well get used to it.

[3]or possibly indicating an undetermined footnote?

Having chosen a basis, we can describe any vector with a tuple of coordinates r^n, so any vector space of dimension N whose scalar field is $\mathbb{F}$ must be isomorphic with $\mathbb{F}^N$. In other words, all vectors can be described by column vectors, but the numbers in the columns will depend on our choice of basis.

But the laws of physics cannot possibly care what basis we choose, so we need ways of obtaining numeric facts about vectors that do not depend on the choice of basis.

3.3 Covectors

Think of a scalar-valued function of a vector. That is, a black box with a single input slot accepting a vector $\mathbf{a}$, and an output hole that gives us back a scalar x (Figure 3.3).

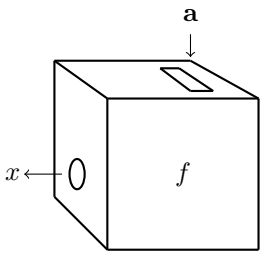


Figure 3.3: Function f with a single slot accepting a vector

More precisely, this machine is a mapping from the vector space to the real numbers. There are infinitely many such mappings and they could be rather complicated. We will restrict ourselves to a simple subset of these mappings.

First, we note that it is possible to define the addition operator on mappings:

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

That is, it could be that inside the box f, there are concealed two boxes g and h. When f receives an input vector $\mathbf{a}$, it passes it to both g and h, and adds their results together to obtain its own result. Note that we haven't yet restricted the complexity of g and h; we have no idea what they do to produce their individual results.

Likewise, it is possible to scale a mapping by a factor x:

$$f(\mathbf{a}) = xg(\mathbf{a})$$

The trick to restricting the complexity of our set of possible mappings is to require that they comply with the rules of a vector space. Not only can they

be added and scaled, but combinations of these operations produce consistent results. But if we do that, then we have also ensured that the set of allowed mappings actually *is* a vector space. Every mapping we care about must be a vector chosen from that space.

Note that we haven't proven that every possible mapping is a vector. We've merely restricted ourselves to only considering a subset of mappings, those that can be scaled and added to find other mappings from the same restricted subset, such that scaling a mapping by 5 is the same as scaling that mapping by 2 and separately by 3 and then adding those two scaled mappings.

If we label the original vector space V then this associated vector space of mappings $V \mapsto \mathbb{R}$ is written as V^* and is called the dual space of V. All we've discovered so far about V also applies to V^*, including the idea of a set of mappings being linearly independent, which means we can select a basis of mappings chosen from V^* and thus construct any mapping from it by scaling and adding the basis mappings. That's quite a leap, so pause to digest it. The moment you discover a set of objects is a vector space, you know you can choose a basis, and then describe anything in that space in terms of a weighted sum of that basis.

We call the mappings taken from V^* *covectors*. The basis covectors are labelled with superscripts $\mathbf{e}^n$ and the coordinates with subscripts f_n, so we can build any covector from the chosen basis:

$$\begin{aligned}\mathbf{f} &= f_1\mathbf{e}^1 + f_2\mathbf{e}^2 + ... + f_n\mathbf{e}^n \\ &= \sum_n f_n\mathbf{e}^n\end{aligned} \tag{3.2}$$

It follows that, just as N-dimensional vectors are isomorphic with columns of N scalars, so too are their associated covectors.

It is sometimes suggested that all vector spaces have a dual space, as if this was some property hiding in the definition of a vector space. But in truth we have conjured the dual space into existence, first by inventing the idea of a mapping $V \mapsto \mathbb{R}$, then by defining operations on those mappings, then by considering the set of all possible mappings, and finally by imposing the rules of vector spaces, which restricts us to a subset of the possible mappings that we named covectors. There is nothing particularly automatic about this. We made it happen by being curious about ways in which vectors might be mapped to scalars.

Another important point to note is that as covectors are vectors, the operation we've been writing as $f(\mathbf{a})$ is in fact much more symmetrical than that notation implies. We combine a vector from V and a covector from V^* and this produces a scalar from $\mathbb{R}$.

So we could equally say that a vector "operates" on a covector to produce the scalar. From the point of view of V^* it is V that is the dual space. There is a more symmetrical notation[4] we can use to make this clear:

$$\langle \mathbf{f}, \mathbf{a} \rangle$$

In this notation, the left and right sides of the operation are from mutually dual spaces, mirror opposites that annihilate one another leaving only a scalar residue.

Although we will mostly think of a covector as a function and a vector as something that can be a parameter to a covector, keep in mind that just as we've thought of a covector as machine that accepts a vector as input, we could just as well think of a vector as a machine that accepts a covector as input.

3.3.1 Connecting the Dual Spaces

We obviously have a lot of freedom when choosing a basis in either V or V^*. What can we usefully do to relate the two sides? We've seen how a covector may be built as a weighted sum of basis covectors $\mathbf{e}^i$ from V^*:

$$\mathbf{f} = \sum_i f_i \mathbf{e}^i$$

And likewise a vector is built as a weighted sum of basis vectors $\mathbf{e}_i$ from V:

$$\mathbf{v} = \sum_i v^i \mathbf{e}_i$$

If we have the vector $\mathbf{v}$ and we want to extract its ith coordinate, v^i, that's a function from a vector in V to a scalar, that is, it's a covector from V^*. We could choose the basis covectors so that the ith basis covector extracts the ith coordinate of the vector passed to it:

$$v^i = \langle \mathbf{e}^i, \mathbf{v} \rangle$$

Equivalently, if we have a covector $\mathbf{f}$ and we want to extract its jth coordinate, f_j, then we need to pass f an input vector chosen from V, and we could choose the basis vectors so that the jth basis vector makes $\mathbf{f}$ produce the jth coordinate of $\mathbf{f}$:

$$f_j = \langle \mathbf{f}, \mathbf{e}_j \rangle$$

[4] Unfortunately there is almost no consistency on notation in this topic; we're just picking one of many possible notations for this.

It doesn't matter which of those two ways we approach this, because either will constrain the other. We can substitute $\mathbf{f}$ expressed as a sum:

$$f_j = \langle \sum_i f_i \mathbf{e}^i, \mathbf{e}_j \rangle$$

and linearity allows us to separately deal with each dimension and sum their results:

$$f_j = \sum_i f_i \langle \mathbf{e}^i, \mathbf{e}_j \rangle$$

But if that's true for *any* $\mathbf{f}$, and not just a coincidence applying to some specific example, then the scalar factor $\langle \mathbf{e}^i, \mathbf{e}_j \rangle$ must be "selecting" just one of the f_i terms, specifically the one where $i = j$, and eliminating all others, or to put it another way:

$$f_j = \sum_i f_i \delta^i{}_j$$

So if the f_j are indeed the components of the covector, discovered by making it act on the basis vectors, we've discovered the relationship that must exist between the dual bases:

$$\langle \mathbf{e}^i, \mathbf{e}_j \rangle = \delta^i{}_j \tag{3.3}$$

So we could choose any basis at all in V, and then definition (3.3) restricts the choice of basis in V^*, or vice versa. Such is the symmetry of this situation, we could instead have let a basis covector $\mathbf{e}^j$ act on a randomly chosen vector $\mathbf{a}$ and require that this give us the jth coordinate of $\mathbf{a}$, and we'd have reached the same conclusion.

From now on we'll assume that this alignment of the dual bases has been performed. That being the case, we can compute the action of a covector on a vector by arithmetic on their coordinates:

$$\begin{aligned} \langle \mathbf{f}, \mathbf{a} \rangle &= \langle \sum_i f_i \mathbf{e}^i, \sum_j a^j \mathbf{e}_j \rangle \\ &= \sum_{ij} f_i a^j \langle \mathbf{e}^i, \mathbf{e}_j \rangle \\ &= \sum_i f_i a^i \end{aligned} \tag{3.4}$$

So the linearity allows us to sum over all combinations of i, j and pull the coordinates outside of the action of the covector on the vector, and the dual bases yield the value 1 where $i = j$ and 0 otherwise, so we just end up with a simple sum over the products of the paired-up coordinates.

In a roundabout way we've discovered the dot product, albeit between a covector and a vector rather than two ordinary vectors. We still haven't introduced any concept of orthonormality, or even orthogonality, between pairs of vectors. Our "coordinate grid" is still not necessarily a lattice of squares, and our dot product is between elements of two different (dual) vector spaces, but we always choose their basis vectors so they are related by a definite requirement, which we can state in two ways:

1. The nth basis covector from V^* can be used to extract the nth coordinate of a vector from V.
2. The nth basis vector from V can be used to extract the nth coordinate of a covector from V^*.

And as a consequence of this (dual) requirement we find that when the nth basis covector acts on the mth basis vector, the result is 1 if $m = n$ and 0 if $m \neq n$.

If we describe our covectors and vectors as sets of coordinates, to make a covector act on a vector we simply perform the dot product between their coordinates. Or equivalently, we write the covector as a single row matrix on the left, and the vector as a single column matrix on the right, and perform matrix multiplication to get a single scalar.

Does this mean we've created a dual link between *every* vector and a corresponding covector? Absolutely not. We've only linked up the basis vectors with the basis covectors. Still, there is an obvious mapping between vectors and covectors: let the covector have exactly the same coordinates as its vector pair, $f_i = v^i$. But that is only one possible mapping out of an infinity of possibilities, as we could blend the coordinates by any weighted sum we like. So connecting the basis covectors with the basis vectors is a start, but still leaves something to be desired.

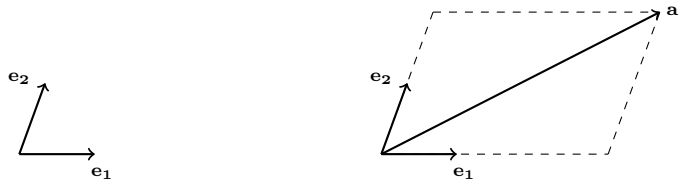
3.3.2 Visualising the dual basis

It may be worth pausing here to see how this result relates to our schoolhouse version of arrows and coordinates and the dot product. In that world-view, the coordinates are just the scaling factors that weight the orthonormal basis vectors to construct a vector, and the dot product of a basis vector $\mathbf{e}_i$ and a given vector $\mathbf{a}$ produces the a_i coordinate of $\mathbf{a}$. This is how we understand the geometric dot product (with $\cos\theta$) to be related to the idea of simply plucking one of the numbers from a column vector.

But what happens if we deny all knowledge of orthogonality? If we choose

any linearly independent basis vectors (Figure 3.4a) we can still sum them to generate any vector in the space (Figure 3.4b).

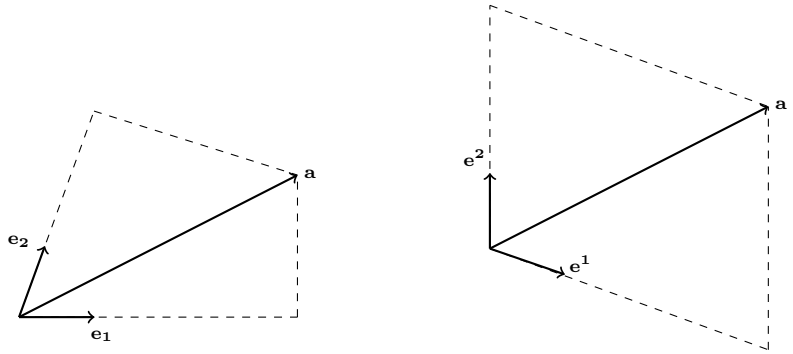
Figure 3.4: Building a vector from any basis



(a) Any old basis $\mathbf{e}_n$ (b) $\mathbf{a} = \mathbf{e}_1 a^1 + \mathbf{e}_2 a^2$

The problem comes when we try to recover the coordinates by projecting $\mathbf{a}$ onto the two basis vectors (Figure 3.5a).

Figure 3.5: Projecting a vector onto a carelessly chosen basis



(a) Projecting $\mathbf{a}$ back onto the basis (b) $\mathbf{a} = a_1\mathbf{e^1} + a_2\mathbf{e^2}$

We can visualise this projection process by drawing lines from the tip of $\mathbf{a}$ so they meet at right angles with the lines extended from the basis vectors. But these imply different coordinates for $\mathbf{a}$ from the ones that we used to build it using the basis $\mathbf{e}_n$.

This raises the question: in what basis are these the coordinates for $\mathbf{a}$? There is such a basis (Figure 3.5b), $\mathbf{e^n}$, and we label the coordinates with subscripts, a_n, so the reconstructed $\mathbf{a}$ is given by:

$$\mathbf{a} = a_1\mathbf{e^1} + a_2\mathbf{e^2}$$

This basis $\mathbf{e}^j$ is related to the original basis $\mathbf{e}_i$ by (3.3). Looking at it geometrically (that is, cheating), when choosing the dual basis vector for a given index, we must choose a vector that is visibly orthogonal to all the other basis vectors,

and this means we will have a severely limited choice, because there can be only one alignment that meets this requirement. Furthermore the magnitude of the vector $\mathbf{e}^i$ is fully determined by the requirement that $\langle \mathbf{e}_i, \mathbf{e}^i \rangle = 1$, as the ratio between the coordinates is already fixed by the choice of alignment.

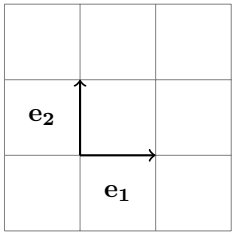
In this visualisation process we have shown the relationship between V and $V*$ by overlaying them on the same diagram, but they are in fact separate vector spaces: elements of V are not elements of $V*$, and vice versa. But the way we have calibrated these two sets of bases to be mutually consistent is exactly the same as the relationship between the bases of V and $V*$.

If the original basis vectors had been orthogonal, the dot product would have produced exactly the same coordinates we'd used to build the vector in the first place, i.e. figures 3.4b, 3.5a and 3.5b would all be identical: a rectangle with the vector as its diagonal. But of course, we haven't yet said precisely what orthogonality means.

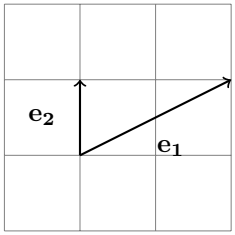
3.3.3 The same ideas in coordinates

We can make this concrete by playing with $\mathbb{R}^2$ as our vector space V, in which case the dual space of covectors V^* contains mappings $\mathbb{R}^2 \mapsto \mathbb{R}$.

Figure 3.6: Basis vectors in $\mathbb{R}^2$



(a) Orthonormality



(b) Awkwardness

With our schoolhouse foreknowledge it would be easy to choose an orthonormal basis in $\mathbb{R}^2$ (Figure 3.6a):

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

But we still haven't defined what orthonormal means, so we'll just choose something awkward (Figure 3.6b):

$$\mathbf{e}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

By the way, it is customary to put the basis vectors in a row matrix, $\begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2\end{bmatrix}$, so they can be matrix-multiplied by a column representation of a vector in V, but that's not what we're doing here. We are giving the definition of each basis vector as a matrix, and the basis vectors are ordinary vectors belonging to V, so they must be column matrices.

What is the corresponding V^* basis, $\mathbf{e}^i$? It has to obey:

$$\langle \mathbf{e}_j, \mathbf{e}^i \rangle = \delta_{ij}$$

Some straightforward equation building and substitution yields:

$$\mathbf{e}^1 = \begin{bmatrix}0.5 & 0\end{bmatrix}, \mathbf{e}^2 = \begin{bmatrix}-0.5 & 1\end{bmatrix}$$

And these being covectors from V^*, they must be row matrices, as shown. Using our V basis we can construct a vector $\mathbf{v}$ from the coordinates $(2, 3)$:

$$\mathbf{v} = 2\mathbf{e}_1 + 3\mathbf{e}_2 = \begin{bmatrix}4\\2\end{bmatrix} + \begin{bmatrix}0\\3\end{bmatrix} = \begin{bmatrix}4\\5\end{bmatrix}$$

What happens if we evaluate the V^* basis covectors against $\mathbf{v}$?

$$\begin{bmatrix}0.5 & 0\end{bmatrix}\begin{bmatrix}4\\5\end{bmatrix} = 2\,,\ \begin{bmatrix}-0.5 & 1\end{bmatrix}\begin{bmatrix}4\\5\end{bmatrix} = 3$$

We get back the correct coordinates. If we'd just transposed the V basis vectors into rows and left-multiplied them, we would have obtained wrong answers: this is precisely the same problem we saw with projecting onto the non-orthogonal basis.

But the utility of these basis covectors is limited to their ability to extract a scalar coordinate from a vector. For example, there is nothing here that generally relates any vector (other than the basis vectors) with a specific covector partner, or anything that relates one vector with another.

3.4 Tensors

Let's upgrade our black box machine so it has two input slots, accepting vectors $\mathbf{a}$ and $\mathbf{b}$ from the same vector space V, but still one output hole that gives us back a scalar x (Figure 3.7).

This is a mapping from pairs of vectors to scalars: $V \times V \mapsto \mathbb{R}$.

Suppose *only to begin with* that the machine had an especially simple inner mechanism: inside the box $\mathbf{h}$, there are two single-slot boxes (covectors) $\mathbf{f}$ and

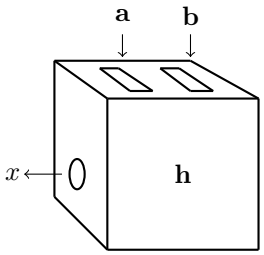


Figure 3.7: Box $\mathbf{h}$ with two slots accepting vectors

$\mathbf{g}$. The machinery inserts input $\mathbf{a}$ into box $\mathbf{f}$, and input $\mathbf{b}$ into box $\mathbf{g}$, to obtain two scalars, which it simply multiplies together to produce its own resultant scalar that falls out of the hole of $\mathbf{h}$. In other words, it's really just two covectors glued together by scalar multiplication, and they act independently on the inputs:

$$h(\mathbf{a}, \mathbf{b}) = \langle \mathbf{f}, \mathbf{a} \rangle \langle \mathbf{g}, \mathbf{b} \rangle$$

In fact this design only accounts for a subset of the machines, but it will serve as an intuitive building block for us to construct all the machines we're interested in. We can write it as $\mathbf{f} \otimes \mathbf{g}$, which is called the *tensor product* of the two covectors.

So sticking with this simplified design to begin with, consider the space of all such machines containing two covectors. We can define addition and scaling on these pairs in a way that satisfies the requirements of a vector space. Addition is easy. Much as we defined addition for covectors by simply adding their results when given the same input vector, we'll add these two-slot machines by adding their results when they act on the same pair of vectors:

$$\begin{aligned} (\mathbf{f} \otimes \mathbf{g} + \mathbf{p} \otimes \mathbf{q})(\mathbf{a}, \mathbf{b}) &= \mathbf{f} \otimes \mathbf{g}(\mathbf{a}, \mathbf{b}) + \mathbf{p} \otimes \mathbf{q}(\mathbf{a}, \mathbf{b}) \\ &= \langle \mathbf{f}, \mathbf{a} \rangle \langle \mathbf{g}, \mathbf{b} \rangle + \langle \mathbf{p}, \mathbf{a} \rangle \langle \mathbf{q}, \mathbf{b} \rangle \end{aligned} \tag{3.5}$$

Scaling by some x is even easier:

$$[x(\mathbf{f} \otimes \mathbf{g})]\,(\mathbf{a}, \mathbf{b}) = x(\mathbf{f} \otimes \mathbf{g})(\mathbf{a}, \mathbf{b}) = x\langle \mathbf{f}, \mathbf{a} \rangle \langle \mathbf{g}, \mathbf{b} \rangle$$

Thus we have defined a vector space, and so these two-slot machines are also vectors. We can form a basis for that space by taking all possible pairs of basis covectors, $\mathbf{e}^i \times \mathbf{e}^j$. If the (co)vector space is N-dimensional, the pair-space will be N^2-dimensional, because it requires N^2 basis machines to span the space. Any two-slot machine can therefore be written as a linear combination (a weighted sum) of all the basis machines:

$$\sum_{ij} M_{ij}(\mathbf{e}^i \otimes \mathbf{e}^j)$$

And therefore to describe any two-slot machine in terms of the basis we will need N^2 numbers, which we can write as M_{ij}. By the way, there's no pressing need to think of it as a matrix, although we sometimes do. The M stands for "machine" in this case. It's just a list of N^2 numbers, labelled with two indices that each can take on N integer values, supplying the weighting for each basis machine.

Inserting two vectors $\mathbf{a}$ and $\mathbf{b}$ into the slots just means:

$$\sum_{ij} M_{ij}\langle \mathbf{e}^i, \mathbf{a}\rangle\langle \mathbf{e}^j, \mathbf{b}\rangle$$

Recall that an expression like $\langle \mathbf{e}^i, \mathbf{a}\rangle$ is the scalar resulting from $\mathbf{e}^i$ acting on $\mathbf{a}$. But as the input vector $\mathbf{a}$ can be described using the same (dual) basis, $a^k\mathbf{e}_k$:

$$\sum_{ijk} M_{ij}\langle \mathbf{e}^i, a^k\mathbf{e}_k\rangle\langle \mathbf{e}^j, \mathbf{b}\rangle$$

and the same for $\mathbf{b}$:

$$\sum_{ijkl} M_{ij}\langle \mathbf{e}^i, a^k\mathbf{e}_k\rangle\langle \mathbf{e}^j, b^l\mathbf{e}_l\rangle$$

Both these substitutions required us to introduce a new summation index, because we are essentially "multiplying out" between the existing expression's summation terms and those of the vector we are substituting. So if the vector space is 2-dimensional, the above is summing $2 \times 2 \times 2 \times 2 = 16$ terms. By linearity:

$$\sum_{ijkl} M_{ij}a^k\langle \mathbf{e}^i, \mathbf{e}_k\rangle b^l\langle \mathbf{e}^j, \mathbf{e}_l\rangle$$

We know the basis covectors acting on the basis vectors have very simple results by definition: $\langle \mathbf{e}^i, \mathbf{e}_k\rangle$ evaluates to 1 if $i = k$, but is 0 otherwise. Likewise $\langle \mathbf{e}^j, \mathbf{e}_l\rangle$ is 1 if $j = l$ but 0 otherwise. Therefore the 12 summation terms where either $i \neq k$ or $j \neq l$ must vanish, leaving only 4 terms where they are equal and the covector-vector interactions are simply replaced with 1. Therefore we can replace k with i and l with j throughout:

$$\sum_{ij} M_{ij}a^i b^j \tag{3.6}$$

So to compute the scalar result we only need the coordinates of the two input vectors and a list of numeric parameters that fully defines how the machine operates, creating summation terms that contribute various weightings of every possible combination of coordinates from the two vectors.

This kind of machine is called a *tensor*.

It will often be the case that $M_{ij} = M_{ji}$, which is quite a lot of redundancy. But this is necessary to ensure that the machine is symmetrical: switching the inputs around does not affect the result. Of course, a machine doesn't have to be defined that way.

3.4.1 Simple tensors

We mentioned at the start that the simplified design (the ordinary product of two scalars obtained by two covectors operating separately on one vector input each) is not powerful enough to describe all these machines, even though we used it to define our basis machines. The simplistic machine is defined by a matrix that can be written as:

$$M_{ij} = f_i g_j$$

In other words, it can be decomposed into two separate columns of numbers. Such a machine is known as a *decomposable*, *elementary* or just *simple* tensor.

One hint as to why it is so limited is that as f_i and g_j provide N values each for an N-dimensional space, that is only $2N$ adjustable parameters, even though M appears to have N^2 independent values. So we aren't allowing the full flexibility of which M is capable.

Of course, if $N = 2$ then $N^2 = 2N = 4$, but even then, there is a restrictive pattern that applies regardless of the dimensions. Viewing M_{ij} as a matrix, every row (labelled by i) would be a scaled version of the numbers in g_j, and every column (labelled by j) would likewise be a scaled version of the numbers in f_i. That is, the rows are all linearly dependent on one another, and so are the columns. This would not be the case if the elements of M were truly independent.[5]

There's a subtlety here though: vector addition and scaling operators are meant to be closed. We've proposed a way of defining simple machines, which is a restricted set of objects, and then we've said that scaling and adding simple machines allows us to discover objects that are not in that simple set, which sounds like we're breaking the rules, reaching outside the initial set.

[5]Can any symmetric matrix be expressed as the product of a row and a column? No. Try to find a row and a column that can be multiplied to produce the identity matrix. The columns are linearly independent, as are the rows.

The resolution to this conundrum is that we are dealing with a general set of machines that can be described by the somewhat misleading notation $V^* \otimes V^*$, which we define as not only the simple machines made of any two covectors $f \otimes g$, but also those machines that are *weighted sums* of one or more simple machines. This broader set includes machines that cannot be decomposed into two covectors. We can however choose a basis from the subset that *can* be decomposed, and we do that because it is that subset for which we are able to directly explain how they operate. And from that basis we can build any machine of the form $V^* \otimes V^*$. Covectors (one-slot machines) are the most basic building block, from which we can make simple (two-slot) machines, from which in turn we can make any machines by linear combination of simple machines.

3.4.2 Any number of slots

Another point to note about these two-slot machines is that although here we focused on $V^* \otimes V^*$, we could instead of chosen $V \otimes V$, in which case the simple machine would have consisted of two vectors, and would have acted on two input covectors (recall how the notation $\langle \mathbf{f}, \mathbf{a} \rangle = \langle \mathbf{a}, \mathbf{f} \rangle$ emphasises symmetry, so we can think of a covector acting on a vector or a vector acting on a covector with no real difference in the result). We can define machines of the form:

$$V \otimes V^* \otimes V \otimes \dots$$

having any number of slots accepting any mixture of vectors and covectors in some specific order. A machine with five slots will be represented by a list of numbers labelled with five indices. The slots that accept vectors (being defined by covectors) will have down indices, and the slots that accept covectors (being defined by vectors) will have up indices. For example, we can say our machine is from the space:

$$V \otimes V^* \otimes V \otimes V \otimes V^*$$

or we can say it is represented by the numerical parameters:

$$M^{i\ \ kl}_{j\ \ m}$$

These convey the same information. Sometimes the indexed parameter notation is used as a compact way to describe the structure of the machine, the only downside being that we have to unnecessarily choose symbols for the indices.[6]

[6]Some authors call this *slot-naming index notation*.

3.4.3 What is a tensor?

These machines are tensors, though as we've seen, the space of machines of a given type is also a vector space, so tensors are vectors.

We have proposed creating a five-slot machine $M^i{}_j{}^{kl}{}_m$, and so it seems entirely proper to treat a 1-slot machine M_i or M^i as part of the same family of objects. We've been calling them covectors and vectors, which is accurate (we had to invent them first in order to build toward tensors), but they are also themselves tensors in this general sense.

Perhaps more surprising, but no less consistent, is the idea that a machine with no slots also belongs to the same family. It's just a scalar value.

Sometimes the type of a tensor is written (u, d) where u tells you how many up indices and d tells you how many down indices it has. So a scalar is a $(0, 0)$-tensor, a vector is a $(1, 0)$-tensor, a covector is a $(0, 1)$-tensor, and we will soon encounter practical uses for $(1, 1)$, $(0, 2)$ and $(2, 0)$-tensors, all of which are also elements of vector spaces.

In summary, everything is seemingly an example of everything else, and yet all are different things.

3.4.4 Contraction

When we insert a vector or covector into a suitable slot of a machine, we are effectively merging two tensors, by "wiring up" them up so that they share an index variable, which makes that variable disappear due to summation over it.

Starting with a five-slot machine $M^i{}_j{}^{kl}{}_m$ (which incidentally is a $(3, 2)$-tensor), we will insert a covector a_k into the middle slot, resulting in a machine N with four slots (a $(2, 2)$-tensor):

$$N^i{}_j{}^l{}_m = \sum_k M^i{}_j{}^{kl}{}_m a_k$$

The index k effectively disappears. This process is more formally regarded as a two stage process. First, we form the tensor product, which is an object with a separately named index for every index of the two source tensors:

$$P^{i\ kl}_{\ j\ mn} = M^i{}_j{}^{kl}{}_m a_n$$

This step doesn't involve any summation. If the vector space is 4-dimensional, P is a list of $4^6 = 4096$ numbers, each being the product of a distinct pair from the $4^5 = 1024$ numbers in M and the 4 numbers in a.

Then we link two of the slots by giving them the same index name and summing over that index (in this case by renaming n to k):

$$N^{i\ \ l}_{j\ m} = \sum_k P^{i\ \ kl}_{j\ \ mk}$$

It's that second step, renaming n to k and summing over k, that is the actual contraction. We could then perform two contractions at once on N:

$$x = \sum_{il} N^{i\ \ l}_{i\ l}$$

Each contraction ties two indices together and eliminates them, so this last double-contraction has eliminated four indices at once, leaving us with a scalar.

3.5 Einstein notation

We have been following a rule where basis vectors are given subscript indices, while vector components are given superscript indices. Then we do the opposite with basis covectors and components. This means that whenever one basis object acts on another:

$$\langle e^i, e_j \rangle$$

they always have opposing index positions. This is mirrored exactly by the way components are allowed to be multiplied. We've found that the dot product is only valid between the coordinates of a vector and a covector, so a product like this inside a summation, where we have repeated the same index variable:

$$a^i b_i$$

is valid, but neither of these is allowed because they imply a dot product between two vectors or two covectors:

$$a^i b^i \ , \ a_i b_i$$

Here's a real example that we'll encounter later:

$$\sum_{\mu\nu\beta\lambda} g_{\mu\nu} Z^{\mu}_{\ \ \beta} \underline{a}^{\beta} Z^{\nu}_{\ \ \lambda} \underline{b}^{\lambda}$$

Every single one of the four indices appears in two places, once up and once down. This immediately tells us that the result is a scalar. A simpler example shows how index variables are not necessarily introduced by summation:

$$b^\mu = \sum_\nu O^\mu{}_\nu a^\nu$$

The ν index is repeated up/down in the way that is characteristic of all summation variables, but μ is introduced on the left to indicate that the expression computes the value of one component of several that represent a vector (we know it's a vector because the index is up).

From these patterns we can deduce something surprising: it is completely unnecessary to write the summation symbol and state what the summation index variables are! If an expression consisting of indexed quantities multiplied together contains exactly two references to the same index, once up and once down, then that index is a summation index.

$$g_{\mu\nu} Z^\mu{}_\beta \underline{a}^\beta Z^\nu{}_\lambda \underline{b}^\lambda = \sum_{\mu\nu\beta\lambda} g_{\mu\nu} Z^\mu{}_\beta \underline{a}^\beta Z^\nu{}_\lambda \underline{b}^\lambda$$

$$b^\mu = O^\mu{}_\nu a^\nu = \sum_\nu O^\mu{}_\nu a^\nu$$

This shorthand applies just as well to basis vectors:

$$\mathbf{a} = a^i \mathbf{e}_i = \sum_i a^i \mathbf{e}_i$$

The recent example of a tensor product cannot be mistaken for an implied summation because there are no repeated indices:

$$P^{i\ kl}_{j\ mn} = M^{i\ kl}_{j\ m} a_n$$

Whereas the contraction example unmistakably sums over k alone:

$$N^{i\ l}_{j\ m} = P^{i\ kl}_{j\ mk}$$

Einstein came up with this abbreviation while developing General Relativity.

3.6 The Inner Product

The most important necessity for a specific machine of the form $V^* \otimes V^*$ is to at last come up with a way to define orthogonality, and the norm (length) of a vector, and thus orthonormality, but also a specific two-way pairing between every vector and a covector partner.

Nominating one such machine for a given vector space, we can call it the *inner product*, and we say that the combination of the vector space and its inner product is an *inner product space*.

As we've incessantly complained, we have so far had no way to judge whether two vectors are orthogonal to one another, even though we know that for arrows or column vectors there are some intuitive ways to choose orthogonal vectors. In our abstract development of the subject there was no such thing as orthogonal or orthonormal.

The choice of an inner product determines which vectors are mutually orthogonal, and also which are normalised (of unit norm). Importantly, it will also pair every vector with a single dual covector (and vice versa).

The notation $(\mathbf{a}, \mathbf{b})$ is sometimes used for the inner product, similar to but deliberately distinct from the $\langle \mathbf{f}, \mathbf{a} \rangle$ notation for the action of a covector on a vector.[7]

The *squared-norm* of a vector $\mathbf{a}$ is $(\mathbf{a}, \mathbf{a})$, so the norm is $\sqrt{(\mathbf{a}, \mathbf{a})}$. In most situations (Newtonian and Quantum) there is a rule that the norm must be *positive definite*, meaning that it is never negative and is only zero for the zero vector. The exception is just about anything involving Einstein, who favours inner products that break this rule in every way.

As it's a two-slot machine, its coordinate representation can be thought of as a matrix or a list of numbers addressed by two indices, and it is called the *metric*. In General Relativity the metric is usually written as g, and its indices are often μ and ν, so applying it to vectors $\mathbf{a}$ and $\mathbf{b}$ will look like this:

$$(\mathbf{a}, \mathbf{b}) = \sum_{\mu\nu} g_{\mu\nu} a^\mu b^\nu$$

Or expressed in matrix multiplication, we put one of the vectors on the left of the matrix, transposed into a row, and one on the right as a column.

$$(\mathbf{a}, \mathbf{b}) = \begin{bmatrix} a^1 & a^2 & a^3 \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \begin{bmatrix} b^1 \\ b^2 \\ b^3 \end{bmatrix}$$

The central square matrix is always symmetric, $g_{\mu\nu} = g_{\nu\mu}$, so the vector inputs can be switched without affecting the result. The square matrix can multiply with the right column first, or with the left row: the order of operations doesn't matter.

[7] Sadly the meanings of these notations are sometimes switched, and they aren't the only notations used.

As we will eventually see, this transposition business will wind up being somewhat messier than simply writing down the summations, in which the symmetry is a lot more obvious. This is one reason why it may not be worth thinking of g (or any other two-slot machine) as a matrix.[8]

Compare it to the familiar dot product, which would be:

$$\mathbf{a} \cdot \mathbf{b} = \sum_{\mu} a^{\mu} b^{\mu}$$

It's the same row/square/column matrix multiplication except the central square matrix is missing, or equivalently it's the identity matrix:

$$(\mathbf{a}, \mathbf{b}) = \begin{bmatrix} a^1 & a^2 & a^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b^1 \\ b^2 \\ b^3 \end{bmatrix}$$

So all the times when you obediently used the dot product to operate on two vectors, you were implicitly setting g to be the Kronecker delta (§1.2.3):

$$g_{\mu\nu} = \delta_{\mu\nu}$$

By definition if the inner product is represented by $\delta_{\mu\nu}$ then the basis vectors are orthonormal. In other words, the inner product being represented by the identity matrix doesn't really tell us anything about the inner product. It tells us that we've chosen a set of basis vectors so that they are orthonormal according to this space's nominated inner product. It is always possible to do this, regardless of what the inner product happens to be.[9]

This means that if we can use a single inner product consistently, we may as well define the basis to be orthonormal according that inner product, which means we will always be able to use the dot product between vectors, and the distinction between vectors and covectors becomes unimportant. We can even define all other multi-slot machines in terms of vectors acting on other vectors (via the inner product). That is, every machine would be of the form $V \otimes V \otimes V \otimes \ldots$, and would accept vectors as inputs. You may wonder if all that care we took to distinguish superscript and subscript indices was a waste of time. Indeed in many contexts, including all Newtonian classical mechanics and quantum mechanics, it is a waste of time.

The exception is General Relativity, where the inner product varies from place to place, becoming a *tensor field*. To describe it, we have to fix the basis somehow, and allow the metric to vary. As a result of this, if you want to understand GR, you need to understand that to get the squared-norm of a vector, you need

[8] Another reason: what if the machine has three or more slots?

[9] Strictly speaking it is always possible for a finite-dimensional inner product space.

to get the equivalent covector so it can act on the original vector. But in many other topics, you can just think of the vector acting on itself.

This also means that the previous examples of what we called awkward basis vectors would in fact be orthonormal if we chose a particular inner product.

3.6.1 No such thing as awkward

Starting with the standard basis in column vectors, we could specify this as the inner product:

$$g_{\mu\nu} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix}$$

We had to pick *some* basis as a starting point or we would not have had a way to write down the inner product in numerical form. We used the standard basis, which is made of very simple one-hot vectors, but we now know that we must not call those vectors orthonormal.

Now we'll move the goalposts and choose a different set of vectors to be our basis, and they will be our usual awkward choice:

$$\mathbf{e}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Let's test our inner product on every possible pairing of these basis vectors:

$$(\mathbf{e}_1, \mathbf{e}_1) = \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = 1$$

$$(\mathbf{e}_1, \mathbf{e}_2) = \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$$

$$(\mathbf{e}_2, \mathbf{e}_1) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = 0$$

$$(\mathbf{e}_2, \mathbf{e}_2) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1$$

The basis is orthonormal if $(\mathbf{e}_i, \mathbf{e}_j) = \delta_{ij}$. Undeniably, these basis vectors are orthonormal. The inner product says so, and who are we to doubt it? Furthermore, the inner product's matrix *when expressed in this basis* is $\delta_{\mu\nu}$.

Every distinct basis provides a different way to describe any vector with coordinates, and likewise any covector. Regardless of the basis they are described in, the result of applying a specific covector to a specific vector will be the same

scalar value. That is, the choice of basis is not a fact about the vector space, but merely a way to describe the vectors in it.

Choosing a different inner product (or equivalently a metric) is not like that. In fact a given physical situation may impose a metric and we have to accept it; it's a fact of nature. So there are two possible reasons why the matrix $g_{\mu\nu}$ might change:

- a change of basis, which will change the coordinates a^μ and b^ν of the two input vectors, as well as the matrix elements $g_{\mu\nu}$, so as to ensure the scalar result of the inner product $(\mathbf{a}, \mathbf{b})$ is unchanged, or
- a change of inner product itself, which will not affect the coordinates of anything else, and thus the scalar result of $(\mathbf{a}, \mathbf{b})$ may be very different.

3.6.2 The metric and its inverse

As always we must be careful not to confuse the coordinate or matrix representation of an object with the object itself. The matrix $g_{\mu\nu}$ is merely a representation in some basis of a machine with two slots awaiting vectors.

We know that internal to that machine, it may be simple (consisting of two covectors, which are single-slot machines ready to operate on the two vectors inserted into the slots, and whose scalar results will be multiplied to get the result) or, more generally, it may be a linear combination of such simple two-slot machines.

Therefore if a single vector is inserted into the first slot, it will be fed into the first slot of one or more simple two-slot machines, which will all feed that same vector into their first covector. This will produce a set of scaling factors that apply to the second covector of each simple two-slot machine. This is in effect a set of single-slot machines being linearly combined into a single-slot machine (a weighted sum of covectors).

And we are left with a new single-slot machine awaiting another vector before it can produce a final scalar. In general a machine with S slots can be nibbled away at sequentially, feeding in the S inputs one at a time, each step reducing by one the capacity of the resultant machine to accept further inputs, until eventually all we are left with is a scalar result.

So as advertised, as well as providing a meaning for orthonormality, the metric also provides a complete pairing between every vector and a corresponding covector. Provide a single vector input to the metric and you will get back *the* covector that is the dual partner to that vector, according to the metric.

It follows that if we find the inverse matrix to $g_{\mu\nu}$, that will act on a covector to reveal its dual vector. Or speaking abstractly, we can build something like the inner product but for getting a scalar from two covectors $(\mathbf{p}, \mathbf{q})$. Again the starting point is the simplest two-slot machine equivalent internally to two

vectors $\mathbf{a}$ and $\mathbf{b}$. Each input covector is combined with one of the vectors and the two scalar results are multiplied together:

$$(\mathbf{p}, \mathbf{q}) = \langle \mathbf{p}, \mathbf{a} \rangle \langle \mathbf{q}, \mathbf{b} \rangle$$

And in general we would allow a linear combination of a basis formed from such simple machines, and of course this would have a matrix representation, because we will choose a vector basis $\mathbf{e}_\mu$, which will determine a covector basis $\mathbf{e}^\mu$:

$$(\mathbf{f}, \mathbf{g}) = \sum_{\mu\nu} g^{\mu\nu} p_\mu q_\nu$$

By our consistent use of subscript and superscript indices, we now have matrix representations for two forms of the inner product:

- $g_{\mu\nu}$ operates on two vectors a^μ and b^ν to produce a scalar, or on one vector a^μ to produce its dual covector p_μ.
- $g^{\mu\nu}$ operates on two covectors p_μ and q_ν to produce a scalar, or on one covector p_μ to produce its dual vector a^μ.

These operations can be performed in two stages. For example, $g_{\mu\nu}$ can first operate on a^μ to produce *something* that is ready to operate on b^ν and finish the job by producing a scalar. That intermediate something, if it can act on a vector to get a scalar, must be a covector with coordinates a_ν, one that is the dual of the vector it was produced from. If the basis happens to be orthonormal, the dual vector and covector will have the same coordinates, $a_\mu = a^\mu$ (because $g_{\mu\nu} = \delta_{\mu\nu}$).

So with reference to the indices being "up" or "down", $g^{\mu\nu}$ is sometimes called the raising operator, because it turns a covector's coordinates (subscript) into vector coordinates (superscript), while $g_{\mu\nu}$ is the lowering operator, working the other way. And described in this way, it is obvious that $g^{\mu\nu}$ undoes the work of $g_{\mu\nu}$, i.e. they are mutually inverse matrices. With our natural bias toward vectors, if $g_{\mu\nu}$ is the metric then $g^{\mu\nu}$ is the *inverse metric*.

When dealing with objects expressed in coordinates we use the convention that vector coordinates are up and covector coordinates are down, and this lets us always take care to ensure that when we sum their products we always pair a vector (up) with a covector (down) using the same index variable. If necessary we can easily raise or lower using the appropriate version of the metric.

3.7 Operators

In this context, an operator $\hat{O}$ is a function that maps from a vector to a vector, $V \mapsto V$. As usual we are particularly interested in linear operators, for which:

$$\hat{O}\mathbf{a} = \hat{O}\left(\sum_n a^n \mathbf{e}_n\right) = \sum_n a^n \hat{O}\mathbf{e}_n$$

Why? Because to completely characterise the behaviour of such an $\hat{O}$ we only need to know its effect on the n basis vectors, which we can capture in a square matrix.

This is a new kind of geometric object. It has a physical meaning that is entirely independent of the choice of basis, even though the matrix elements will certainly be different depending on the basis. So just as we can speak of the same vector being represented by different columns of coordinates depending on the choice of basis, we can also speak of the same operator being represented by matrices of different elements depending on the choice of basis.

When we label a matrix with indices, M_{ij}, we have choice: should they be subscript or superscript? This was straightforward when we introduced the inner product, because we conceived of it as two covectors tied together by multiplication: $V^* \otimes V^*$, and so their coordinate representations must have subscripts, and hence the metric has subscript indices. Likewise the inverse metric is like two vectors, $V \otimes V$, and so has superscript indices.

In both those cases, we found that as well as acting as a mapping from two inputs to a scalar, the same object could map a single input to its dual partner.

Here we want a machine that maps a vector to a vector. As a hunch, consider the set of $V \otimes V^*$ machines. Such a machine will be a linear combination of simple machines made of a vector and a covector, $\mathbf{a} \otimes \mathbf{f}$:

$$\hat{O} = \sum_{\mu\nu} O^\mu{}_\nu \left(\mathbf{e}_\mu \otimes \mathbf{e}^\nu\right) \tag{3.7}$$

So the operator is represented (in the current basis) by a matrix with one super and one sub index, being the sum of a set of matrices whose elements are the product of a vector coordinate and a covector coordinate. To find the coordinates of the resultant vector $\mathbf{b}$ obtained from $\hat{O}\mathbf{a}$:

$$b^\mu = \sum_\nu O^\mu{}_\nu a^\nu$$

There are some operators that have the same coordinates in all bases. The 0-matrix, which maps all vectors to the 0-vector, has all its elements 0. The identity operator, which has no effect on its input, is $\delta_{\mu\nu}$ in all bases.

A unitary operator is one that is equivalent to a combination of rotation and reflection. One handy fact about such an operator is that if we have its matrix representation (in any basis), we can find the inverse simply by transposing it:

$$(O^\intercal)O = I$$

We can think of a vector as pointing in a direction in space, and this being an aspect of its geometrical nature. How can we similarly characterise operators?

3.8 Eigenvectors and Eigenvalues

An operator that performs only scaling is isotropic, treating all directions equivalently. We can (in somewhat woolly terms) think of it as sphere-shaped.

But some operators are biased with regard to direction. To characterise the behaviour of an operator we can consider those vectors which are scaled by it without their direction being altered (the scaling may be negative, leaving the vector pointing the opposite direction; as long as the resulting vector is co-linear with the input vector, that's insignificant enough.) Such vectors are called the *eigenvectors* of the operator, and the scaling value associated with each eigenvector is called an *eigenvalue*. Note that the zero vector is not considered a candidate for an eigenvector. It has no direction, making it impossible to say whether the operator has changed its direction. Also it goes from length zero to length zero under any operator, so any scalar could be the eigenvalue, meaning that the eigenvalue is undefined.

For operators that perform a general isotropic scaling, all input vectors are eigenvectors: all inputs get only scaled, and always by the same eigenvalue.

With vectors in the plane, when the operator is a pure rotation, e.g. by a right-angle anti-clockwise:

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

or clockwise:

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

every vector changes direction by the same angle, and that means there are no eigenvectors.

More interestingly, there are operators for which only some vectors are eigenvectors. In three dimensions the rotation has an axis, along which all vectors are eigenvectors, and any vectors not on that axis are not eigenvectors. Curiously, even though these eigenvectors don't exist in the planar case, we can find *complex* eigenvalues for them by supposing that such eigenvectors exist, which is weird.

Consider a reflection (call it M for mirror):

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

If we take the first coordinate to be horizontal and the second vertical, this flips the input vector to point up rather than down, or vice versa. So it seems that all vectors have their direction changed and are not eigenvectors, but there exceptions: vectors that lie on the horizontal axis and have no vertical component will be unaffected, i.e. they will be eigenvectors with eigenvalue 1. Also vectors that lie on the vertical axis will have their direction changed, but to the exact opposite direction (their alignment does not change), which is the same as being scaled by -1, and so these too are eigenvectors, but with eigenvalue -1.

So within the space of input vectors, there is a subspace (the *eigenspace*) of eigenvectors, and M has an intrinsic orientation, as there is a particular line around which reflection occurs.

We can find the eigenvectors given a matrix representation M of an operator acting on a vector represented by a column matrix v. If v is an eigenvector, and x is the corresponding eigenvalue, what we mean by that is:

$$Mv = xv$$

For the vector v, multiplying it by the matrix M is the same as multiplying it by the ordinary number x. So with trivial algebra:

$$Mv - xv = 0$$

This is fine because Mv and xv are column matrices, and here 0 is the column matrix filled with zeros.

Pulling out v as a factor (not quite as trivial):

$$(M - xI)v = 0$$

We have to leave behind the identity matrix I in place of v because we need the matrix equivalent of the ordinary number x, so we can subtract it from M.

Having split it into two factors whose product is zero, we know that either one or both of the factors must be zero. We already realised we aren't interested in the case where v is the zero vector because its eigenvalue will be undefined (it could be anything). So we refuse to accept 0 as an eigenvector. Therefore v is not the zero vector, and so the matrix $M - xI$ must be such that it is able to transform some non-zero vector v into the zero vector. This means it has no inverse, as there is no way to recover the direction of the original vector

if we've sent it to 0 (the zero vector has no direction, or has all directions, so direction is a meaningless concept for it.)

Restricting ourselves to a two dimensional vector space, one way to picture the effect of a matrix is to think of it acting on a unit square (where the matrix is 2×2) and asking what the area of the resulting parallelogram will be. If it is not zero, every point in the original square has a unique point in the parallelogram and vice versa: the matrix is invertible. If the area is zero, the points of the original square have been crammed onto a line of 1 dimension, so we have destroyed the information about where they came from in 2 dimensions. All input vectors end up pointing in the same direction, and are only distinguished by length. No linear transform will be able to spread them back out into the correct different directions: the matrix is not invertible.

The area of that parallelogram (or in higher dimensions, the volume of an n-parallelepiped) is called the determinant of the matrix, $\det M$. If it's zero, the matrix is not invertible. And therefore, if:

$$\det (M - xI) = 0$$

then we definitely have some eigenvalues. The determinant can be expanded out into a polynomial expression in x (there are various methods; a popular one is to get a computer to do it) and then solved by factoring to find all the x values that make one of the factors 0. We can then plug those x values back into:

$$(M - xI)v = 0$$

one at a time, and solve to find the corresponding v. Thankfully all this can be mechanised.

3.8.1 Symmetric Matrices

A matrix where $M^\intercal = M$, or $M_{ij} = M_{ji}$ is of particular interest. We've seen how it can be used to define an operation on two vectors that produces a scalar, such that the operation is symmetric (commutative).

Also it is possible to select from its eigenvectors a set that are orthogonal and completely span the vector space. That is, in an N dimensional space, they perform a scaling in all N available orthogonal directions, stretching or squishing. We need to be clear about what we mean by orthogonal, of course: this is a matrix in isolation from any defined geometric context, so we're talking about column vectors in the standard basis.

This is hugely important in Quantum Mechanics (§4), albeit with some modifications for complex numbers.

3.9 Change of Basis

We have now accumulated a menagerie of geometric objects that can be described numerically relative to a chosen basis. We now turn to the problem of what happens to the those numerical representations when we choose a different set of basis vectors.

Often in discussions of transformations a tick is added to existing symbols to indicate the transformed version of that symbol, e.g. x becomes x' (pronounced "x prime"). But this clashes horribly with superscripts. So here we will use underlining, e.g. the vector $\mathbf{a}$ may have a first representation in coordinates a^i and also a transformed representation $\underline{a}^i$.

3.9.1 Effect of change of basis on vectors

A vector $\mathbf{v}$ in some basis can be expressed in coordinates as a column matrix v:

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

Note that, unlike some prior examples, we are not saying the vector *is* the column matrix. We are saying nothing at all about the nature of the vectors, except that they are elements of some vector space. We are describing in numbers a vector of unknown type taken from some vector space, which we can only do because we have chosen a basis, e:

$$e = \begin{bmatrix} \mathbf{e_1} & \mathbf{e_2} \end{bmatrix}$$

Matrix multiplication builds the vector:

$$\mathbf{v} = ev = 3\mathbf{e_1} + 4\mathbf{e_2}$$

We can create a matrix that will double the length of the basis vectors:

$$Z = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

By the rules of matrix multiplication the bases must be on the left.

$$\underline{e} = eZ = \begin{bmatrix} 2\mathbf{e_1} & 2\mathbf{e_2} \end{bmatrix}$$

What coordinates would $\mathbf{v}$ have in this new basis $\underline{e}$? Intuitively the coordinates need to be halved to refer to the same vector. So we need the inverse of Z, written as Z^{-1}, which shrinks the coordinates:

$$Z^{-1} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

And so our vector's coordinates become:

$$\underline{v} = Z^{-1}v = \begin{bmatrix} 1.5 \\ 2 \end{bmatrix}$$

This is the same vector as before, just in different coordinates:

$$\mathbf{v} = ev = \underline{e}\,\underline{v}$$

We say that vectors are *contravariant* under a change of basis, because their coordinates are transformed by the inverse of the matrix used to transform the basis.

This shows how important it is to be clear about how we're defining the adjustment made to the basis. A transformation Z applied to the basis vectors implies a transformation Z^{-1} on the coordinates of vectors. So when we talk about similar effects on other objects, to avoid confusion we'll consistently start with Z, the transformation that updates the basis.

3.9.2 Effect of change of basis on covectors

A covector can be thought of as a yardstick or a measuring device. By evaluating it with some vector parameter, you are taking a measurement of that vector in the form of an ordinary number, something that is independent of any choice of basis and thus independent of the representation in coordinates.

The ordinary vector $\mathbf{x}$ is represented by a column matrix x of coordinates in our initial basis. As we're updating the basis with Z, therefore we must left-multiply the coordinates of the vector by Z^{-1} to get it in the new coordinate system:

$$\underline{x} = Z^{-1}x$$

But for the covector $\mathbf{f}$, which is a row matrix f, we right-multiply by the original Z (not the inverse):

$$\underline{f} = fZ$$

This ensures that the two representations give us exactly the same numerical result when we evaluate the function on the vector, regardless of the basis:

$$\langle \mathbf{f}, \mathbf{x} \rangle = \underline{f}\,\underline{x} = (fZ)(Z^{-1}x) = f(ZZ^{-1})x = fx$$

The scalar $\langle \mathbf{f}, \mathbf{x} \rangle$ is a basis-independent physical fact. To balance the effect of a coordinate transformation, whatever we do to the coordinates of our vectors, we must do the inverse to the coordinates of our covectors.

3.9.3 The inner product under change of basis

The inner product g operates on two vectors. We've written it in matrix multiplication form by putting the transpose $a^\intercal$ of the first vector on the left, so it becomes a single row matrix, with the second vector on the right as a column matrix b:

$$(\mathbf{a}, \mathbf{b}) = a^\intercal g\, b = \sum_{\mu\nu} a^\mu g_{\mu\nu} b^\nu$$

If we transform the basis by Z, we know the coordinate columns a and b for the two vectors must transform by Z^{-1} to become $\underline{a}$ and $\underline{b}$. To produce the same result, the metric square matrix g must be replaced with a new matrix $\underline{g}$.

$$(\mathbf{a}, \mathbf{b}) = \underline{a}^\intercal \underline{g}\, \underline{b} = a^\intercal g\, b$$

So in practical terms, we need a new two-slot machine $\underline{g}$ that hides the original g machine in a compatibility layer, which "un-transforms" the two input vectors before passing them on to its concealed inner g.

To get b back from $\underline{b}$ is easy, because we just left multiply by Z to cancel out the Z^{-1}:

$$b = Z(\underline{b}) = Z(Z^{-1}b) = (Z\,Z^{-1})b = b$$

To get a from $\underline{a}$ would be the same, except we want to get $a^\intercal$ from $\underline{a}^\intercal$, so instead we right-multiply by $Z^\intercal$, which looks awful but is really the same thing with everything transposed and with the order flipped:

$$a^\intercal = (\underline{a}^\intercal)Z^\intercal = (a^\intercal (Z^{-1})^\intercal)Z^\intercal = a((Z^{-1})^\intercal)Z^\intercal) = a$$

So un-transforming the left and right sides means we can use the unmodified g:

$$(\mathbf{a}, \mathbf{b}) = (\underline{a}^\intercal Z^\intercal) g (Z\underline{b})$$

But associativity means we can bracket the middle square matrix separately:

$$(\mathbf{a}, \mathbf{b}) = \underline{a}^\intercal (Z^\intercal g Z) \underline{b}$$

And so:

$$\underline{g} = Z^\intercal g Z$$

By far the ugliest part of this was all the transposition on the left side. But we can avoid matrix notation entirely and use summation. We know that in the original basis:

$$(\mathbf{a}, \mathbf{b}) = \sum_{\mu\nu} a^\mu g_{\mu\nu} b^\nu$$

This clarifies that we're just adding a lot of products: a^μ, $g_{\mu\nu}$ and b^ν are all just ordinary numbers that we're multiplying so although we've written them in the same order they appear in the matrix multiplication, that is entirely unnecessary here.

We've seen how a machine like g can be defined as a linear combination of a set of basis machines that span the space of all possible machines, each basis machine being formed by choosing a distinct pairing of the basis covectors:

$$\sum_{\mu\nu} g_{\mu\nu} (\mathbf{e}^\mu \otimes \mathbf{e}^\nu)$$

And how that machine acts on two vectors like so:

$$\sum_{\mu\nu} g_{\mu\nu} \langle \mathbf{e}^\mu, \mathbf{a} \rangle \langle \mathbf{e}^\nu, \mathbf{b} \rangle$$

But this time we'll build the vectors $\mathbf{a}$ and $\mathbf{b}$ from their coordinates in the transformed basis, $\underline{a}^i$ and $\underline{b}^i$.

The basis is transformed by Z, and so the coordinates of the vectors will have been transformed by Z^{-1}, and thus to recover the coordinates in the original basis we'll have to use Z:

$$\mathbf{a} = \sum_{\alpha\beta} Z^\alpha{}_\beta \underline{a}^\beta \mathbf{e}_\alpha \, , \; \mathbf{b} = \sum_{\kappa\lambda} Z^\kappa{}_\lambda \underline{b}^\lambda \mathbf{e}_\kappa$$

By the usual substitution and linearity jiggling (yes, that's six indices, so $2^6 = 64$ summation terms if the space is 2-dimensional):

$$\sum_{\mu\nu\alpha\beta\kappa\lambda} g_{\mu\nu} Z^{\alpha}{}_{\beta} \underline{a}^{\beta} Z^{\kappa}{}_{\lambda} \underline{b}^{\lambda} \langle \mathbf{e}^{\mu}, \mathbf{e}_{\alpha} \rangle \langle \mathbf{e}^{\nu}, \mathbf{e}_{\kappa} \rangle$$

But also as usual, most of the terms vanish because that's how basis covectors operate on basis vectors. We can replace α with μ and κ with ν, eliminating two indices:

$$\sum_{\mu\nu\beta\lambda} g_{\mu\nu} Z^{\mu}{}_{\beta} \underline{a}^{\beta} Z^{\nu}{}_{\lambda} \underline{b}^{\lambda}$$

Merely reordering and grouping so that the input vector coordinates are last:

$$\sum_{\beta\lambda\mu\nu} \left[g_{\mu\nu} Z^{\mu}{}_{\beta} Z^{\nu}{}_{\lambda} \right] \underline{a}^{\beta} \underline{b}^{\lambda}$$

This is an example of a self-contained set of summed terms that can be extracted, by defining:

$$\underline{g}_{\beta\lambda} = \sum_{\mu\nu} g_{\mu\nu} Z^{\mu}{}_{\beta} Z^{\nu}{}_{\lambda}$$

Which is the rule for transforming g to account for a basis transformation Z, giving us two ways to calculate the same inner product value:

$$(\mathbf{a}, \mathbf{b}) = \sum_{\beta\lambda} \underline{g}_{\beta\lambda} \underline{a}^{\beta} \underline{b}^{\lambda} = \sum_{\mu\nu} g_{\mu\nu} a^{\mu} b^{\nu}$$

3.9.4 Linear operators under change of basis

We've just seen how to transform a matrix, g into $\underline{g}$, so now we know how to transform matrices, nothing to see here, move along, right?

A linear operator is a mapping $V \mapsto V$, whereas the inner product maps $V \mapsto V^*$. Among other observations, we saw that there is always a basis in which the inner product is represented by $g_{\mu\nu} = \delta_{\mu\nu}$, but if a linear operator is represented by $\delta_{\mu\nu}$ then it maps every vector to itself, and that means it must be represented by $\delta_{\mu\nu}$ *regardless of the basis*. So they are not the same.

3.10 Displacement in a scalar field

3.10.1 Temperature as a function of position

We often have a scalar field (in the physics sense), that is, a scalar-valued function of position in space. An example is a survey of the temperature at points

on a tabletop. This is highly unlikely to be a linear function of position. In one region of the table there is a hot cup of tea, and in another there is an ice bucket of champagne. The temperature is a complex, messy function of position.

We can then ask what by what amount does the temperature change if we move from our current position to another spot nearby. This temperature change is given by a scalar-valued function of the displacement vector (like a covector). Clearly this function has a different definition at each location. But to be accurate over any distance it would need to be arbitrarily complicated (unlike a covector)

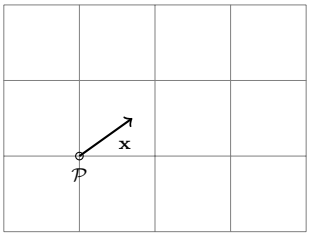


Figure 3.8: A small displacement from $\mathcal{P}$.

The position of a point on the tabletop (Figure 3.8) is given by a position vector and the temperature by the function T. We start at the point $\mathcal{P}$, reached by the position vector $\mathbf{p}$, where the temperature is $T(\mathbf{p})$, and we want to know by how much the temperature will change if we move by a small displacement $\mathbf{x}$.

The answer is given by the function $dT(\mathbf{x})$, which will be a different function depending on where $\mathcal{P}$ is. We can sort of cheat and define it in terms of T with perfect accuracy:

$$dT(\mathbf{x}) = T(\mathbf{p} + \mathbf{x}) - T(\mathbf{p})$$

But that would be overkill. All we really need is a linear approximation that is only precisely accurate at $\mathcal{P}$, where it has the value zero, but will change linearly with increasing distance (and may diverge from the truth).

That is, for any scalar constant k:

$$dT(k\mathbf{x}) = k\, dT(\mathbf{x})$$

As k shrinks, the error vanishes:

$$dT(\mathbf{x}) = \lim_{k \to 0} \frac{T(\mathbf{p} + k\mathbf{x}) - T(\mathbf{p})}{k} \tag{3.8}$$

Having chosen a direction, the dT function's value will be proportional to the distance moved, which is to say, the magnitude of the supplied displacement

vector. The smaller the displacement, the smaller the value, but also the smaller the error. For large displacement it may be wildly wrong; it doesn't carry enough information about the shape of the temperature map contained in T to reproduce it perfectly. It only knows something about how T changes in the immediate vicinity of $\mathcal{P}$, but that's enough, because if T is a complicated function then there will be a different dT function associated with each point in space, and that variation will encode the underlying shape of T.

A function that maps vectors to scalars in this simple linear way is clearly a covector, $\mathbf{d}$. Given a covector basis $\mathbf{e}^n$, there will be a set of coordinates d_n:

$$dT(\mathbf{x}) = \mathbf{d} = \sum_n \mathbf{e}^n d_n \tag{3.9}$$

For every covector there is a dual vector, from the vector space that we use to describe the displacement x, that we could use to convey the same information as $\mathbf{d}$. If we call that vector $\mathbf{s}$ (short for "slope", for reasons that may become clear), we know how to convert covector coordinates into vector coordinates with the inverse metric $g^{\mu\nu}$:

$$\begin{aligned} \mathbf{s} &= \sum_\mu \left(\sum_\nu g^{\mu\nu} d_\nu \right) \mathbf{e}_\mu \\ &= \sum_\mu s_\mu \mathbf{e}_\mu \end{aligned} \tag{3.10}$$

Of course just as the function df (a.k.a. the covector $\mathbf{d}$) is different at each point in the physical space, so too is the corresponding vector $\mathbf{s}$. It's a vector field. The official name for $\mathbf{s}$ is the *gradient*[10] of $T(x)$.

3.10.2 Uniform gradient in orthonormal basis

To drastically simplify, we'll consider a scalar field that has the same gradient everywhere in space. This means that the temperature function of absolute position $T(\mathbf{p})$ depends in a simple way on the position. We can define it in familiar way by using the standard basis, that is, orthonormal basis vectors $\mathbf{e}_n$, which are shown in Figure 3.9 overlaid on dashed lines of equal temperature.

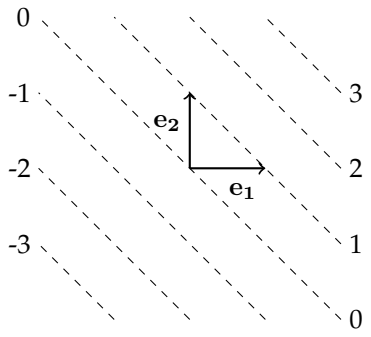
The temperature as a function of the position coordinates p^n is given by:

$$T(\mathbf{p}) = p^1 + p^2 \tag{3.11}$$

The temperature is simply the sum of the coordinates, and so at the origin the temperature is 0, likewise at coordinates $(1, -1), (2, -2), (-1, 1), (-2, 2)$ and so

[10] We can't reasonably call it $\mathbf{g}$, as that is the metric (blame Gauss.)

Figure 3.9: Simplified scalar field



on, which is to say that there is a diagonal line along which T is 0. Similarly there is a parallel line along which T is 1, another where it is -1 and so on. These lines are *isolines* or *contours*[11]

The simplification arsing from a scalar field whose contours are evenly spaced parallel lines is that although T changes with position, there is no variation in *how it changes* with position. This means that the function dT for the change made by a small displacement $\mathbf{x}$ is structurally identical to (3.11):

$$dT(\mathbf{x}) = x^1 + x^2 \tag{3.12}$$

(As always in this topic, x^2 means the second coordinate of a column vector, not the square of a variable x.) We can express this function in matrix multiplication:

$$T(\mathbf{x}) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \end{bmatrix} \tag{3.13}$$

That is, it's the action of a covector with coordinates $(1, 1)$, which we called $\mathbf{d}$ in (3.9):

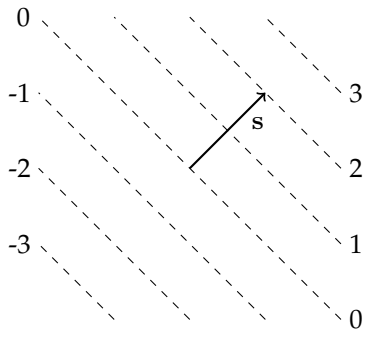
$$T(\mathbf{x}) = \langle \mathbf{d}, \mathbf{x} \rangle$$

The dual of $\mathbf{d}$ is the vector we called $\mathbf{s}$ in (3.10), and because we're in the standard orthonormal basis the coordinates are the same, $d_n = s^n$. This is the gradient vector and can be legitimately drawn in physical space as a vector (Figure 3.10).

Note how it is perpendicular to the contours. It points in the direction of "steepest increase" of the field. Also its magnitude is significant: if you move 1 unit

[11]Strangely they are sometimes called "contour lines" even if they are curved paths.

Figure 3.10: Gradient vector



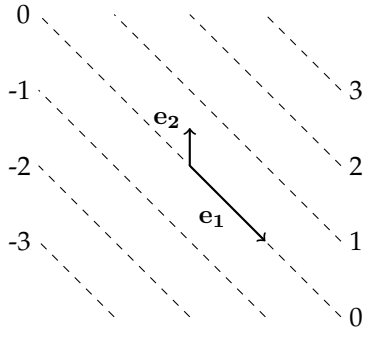
of distance along the direction of $\mathbf{s}$, the field will increase by $\|\mathbf{s}\|$, which in this case is $\sqrt{2}$.

(The diagram is possibly confusing on this point, because it actually shows that a displacement by $\mathbf{s}$, i.e. moving along the direction of s by distance $\|\mathbf{s}\|$, will increase T by 2, which is the same thing.)

3.10.3 Uniform gradient in a non-orthonormal basis

How can we be sure that this description wasn't affected by our use of the standard basis?

Figure 3.11: Simplified scalar field in another basis



Let's rewind back to the start and pick a different basis (Figure 3.11). We've changed $\mathbf{e}_1$ to point diagonally downwards and to the right, so in terms of the standard basis it would have coordinates $(1, -1)$. We've also made $\mathbf{e}_2$ half its previous length.

The significant thing about this (apart from the abandoning of both orthogonality and equal lengths) is that $\mathbf{e}_1$ is parallel to the contours. This means that

a displacement that adjusts only the first coordinate will not affect the value of T, because it just moves along the current contour. But a displacement in the second coordinate will have half the effect it had before. So we expect:

$$dT(\mathbf{x}) = \frac{1}{2}x^2 \tag{3.14}$$

(Again, that is not "x squared"! It's the second coordinate of $\mathbf{x}$.) Or in matrix multiplication it's:

$$T(\mathbf{x}) = \begin{bmatrix} 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \end{bmatrix} \tag{3.15}$$

That is, it's the action of a covector $\mathbf{d}$ with coordinates $(0, \frac{1}{2})$:

$$T(\mathbf{x}) = \langle \mathbf{d}, \mathbf{x} \rangle$$

As always, a covector $\mathbf{d}$ must have a dual vector $\mathbf{s}$, but this time we can't simply transpose the row of coordinates d_n into a column of s^n. The coordinates of the dual vector/covector pair are different in a non-orthonormal basis. We can use the inverse metric (the "raising" metric) to convert the coordinates, if we can figure out what it is. The shortcut is to express our basis vectors as coordinates in terms of the standard orthonormal basis:

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix}$$

and then the lowering metric's elements can be calculated by the dot product[12]:

$$\begin{aligned} g_{\mu\nu} &= \mathbf{e}_\mu \cdot \mathbf{e}_\nu \\ &= \begin{bmatrix} 2 & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{4} \end{bmatrix} \end{aligned} \tag{3.16}$$

The raising metric is the inverse of that:

$$g^{\mu\nu} = \begin{bmatrix} 1 & 2 \\ 2 & 8 \end{bmatrix} \tag{3.17}$$

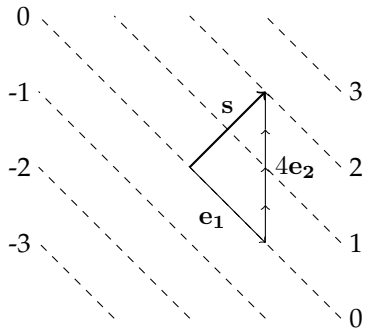
So we can apply that to our transposed d_n coordinates to get s^n coordinates:

$$\begin{bmatrix} 1 & 2 \\ 2 & 8 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

[12]This is only allowed because we're temporarily back in the standard orthonormal basis!

And once again, these being the coordinates of an ordinary vector, we can legitimately plot that vector on the same diagram of the field's contours:

Figure 3.12: Gradient vector in other basis



For absolute clarity we've spelled out the scaling and addition of the basis vectors, to show that $\mathbf{s}$ is equivalent to adding 4 lots of $\mathbf{e}_2$ on to the end of $\mathbf{e}_1$, just as the coordinates of $\mathbf{s}$ tell us to do. But the key point is that this is physically the same vector we found when working in the standard orthonormal basis (Figure 3.10). It is described by different coordinates, of course, but it's the same vector, pointing in the direction of steepest increase. It even has the same magnitude $\|\mathbf{s}\|$, which is the square root of the inner product of $\mathbf{s}$ with itself. The inner product will use the lowering metric to convert one of those $\mathbf{s}$ inputs to $\mathbf{d}$ so it can act on the other $\mathbf{s}$, ensuring a consistent result.

Similarly, given $\mathbf{s}$ and some other vector $\mathbf{x}$ representing a displacement, we can directly obtain the change in T due to that displacement using the inner product $(\mathbf{s}, \mathbf{x})$ between them:

$$\begin{aligned} dT &= (\mathbf{s}, \mathbf{x}) \\ &= \sum_\nu \left(\sum_\mu g_{\mu\nu} s^\mu \right) x^\nu \\ &= \sum_\nu d_\nu x^\nu \end{aligned} \tag{3.18}$$

3.10.4 Smoothly varying gradients

Generalising this to the more complicated kinds of T scalar field, the scalar-valued function $dT(\mathbf{x})$, which tells us how T changes due to a small displacement $\mathbf{x}$, may have a different definition at each point in space. But we can retain the simplification that we only need to know the linear approximation of that function at each point.

The displacement will be described by a number in each dimension: a relative coordinate change associated with each position basis vector. If we only allow one coordinate, x^i, to vary and leave the others constant, dT becomes a scalar-valued function of a single variable, x^i, which if plotted would simply be a line through the origin (i.e. it is 0 when its parameter is 0). It is therefore characterised entirely by a single number, its gradient, the number by which it multiplies its input.

The function T is of course not necessarily linear in x^i. Following the Maclaurin method (§2.4), we assume that whatever the function is, it can be expressed as a polynomial series[13] of the form $d_n x^n$. The first term ($n = 0$, $x^0 = 1$) is just a constant: the value of T at the current position. We will be subtracting this from the total, because we want to know how T changes as we move away from that position. The terms $n > 1$ involve x^2, x^3 and so on. As x becomes arbitrarily small, these terms plummet in significance very rapidly. So that leaves just the term $d_1 x$, and d_1 is the only number we need to describe the tangent line to the curve of the function at the current position. We can therefore discard its subscript label, and replace it with i, as it's the sole outcome of this analysis when varying the ith component of $\mathbf{x}$:

$$\frac{\partial T(\mathbf{x})}{\partial x^i} = d_i$$

We carry out this analysis i times to get the linear gradient number for all i components, and thanks to linearity we can simply add them to get the total change in F:

$$dT(\mathbf{x}) = \sum_i \frac{\partial T(\mathbf{x})}{\partial x^i} = \sum_i d_i x^i = \langle \mathbf{d}, \mathbf{x} \rangle$$

Once again it's just the covector $\mathbf{d}$ described by coordinates d_i, acting on the displacement vector $\mathbf{x}$ described by coordinates x^i. Then to obtain the components of the gradient vector, which is the dual of d, we apply the raising metric $g^{\mu\nu}$ to d.

The trivial example of the linear field with constant gradient $T(\mathbf{x}) = x^1 + x^2$ was therefore in fact fully illustrative, because if the function had any terms that weren't linear, we would have just ignored them anyway.

The way this is usually described, the operator that obtains the field of gradient vectors (the $\mathbf{s}$ associated with each point in space) from a field of scalars $T(\mathbf{p})$ is $\nabla T(\mathbf{p})$, the ∇ being pronounced "del". This is very often conflated with the *total derivative*, which is the somewhat confusing name for the covector $\mathbf{d}$ associated with each point in space. The reason they are often casually conflated is

[13] Warning! In this context we're dealing with scalar variables and x^n briefly means "x to the power of n", not the nth component of $\mathbf{x}$.

because if the basis vectors are orthonormal then they have the same coordinates. A careful author will be clear about this requirement when claiming that "the dot product" (rather than the properly defined inner product) can be used to get the *directional derivative* from the gradient vector and the unit vector in the direction of interest.

3.10.5 Sources of confusion

A major potential source of confusion in this area is the distinction between the total derivative, a covector, and the gradient, a vector, and how and why they might be affected by a change of basis. As we've seen, they are *not* affected by a change of basis. Vectors and covectors never are. Their coordinates are affected, of course.

Another source of confusion is the way we think of the gradient as the rate of change of the scalar field, computed with respect to a change in position. This means that a physical distance appears on the bottom of a fraction, and therefore if we increase the size of our unit of distance, the number on the bottom of the fraction decreases, which increases the overall value of the fraction. This reminds us of covariance; is this the source of some inherently covector-like nature in gradients? No. As we've seen, the gradient is actually a vector, and so its coordinates transform contravariantly. The total derivative is a covector that maps a displacement vector to a change in the scalar field, so its coordinates transform covariantly. These two representations actually convey the same information (even in the form of the same set of numbers if the basis is orthonormal), so there is nothing inherently covariant about that information.

3.11 Complex vector spaces

We've mentioned that a vector space has an associated set of scalars, and this is often $\mathbb{R}$, but it may instead be $\mathbb{C}$, meaning that vectors can be scaled by a complex number, and if expressed as a column of coordinates, those coordinates may be complex.

Even the simplest non-trivial example of such a space, $\mathbb{C}^2$, is not directly imaginable, because although each vector requires two coordinates, each of those is a complex number incorporating a real and imaginary part, so each vector requires four real numbers to describe it, and so $\mathbb{C}^2$ is isomorphic with $\mathbb{R}^4$, which is impossible to visualise directly.

Even so, concepts applicable to real vector spaces also work for complex, although with some modifications. Linear independence of a set of N vectors still means that no one of them can be recreated via a linear combination of the others. A mapping from a vector to a scalar, where the scalar is a complex number, is still called a covector. All the related concepts work the same.

3.11.1 Complex dual space

A basis of linearly independent vectors can be chosen, and a dual basis in the covectors is then defined by (3.3):

$$\langle \mathbf{e}^i, \mathbf{e}_j \rangle = \delta^i{}_j$$

For example, regarding column vectors from $\mathbb{C}^2$, these are linearly independent:

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ i \end{bmatrix}$$

We gleaned from our experiments with one-hot basis vectors in $\mathbb{R}^2$ that $\mathbf{e}^1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$, to ensure that $\langle \mathbf{e}^1, \mathbf{e}_1 \rangle = 1$. By leaning on our schoolhouse assumption of orthonormality based on the dot product, we observed that if the basis vectors are orthonormal, each basis covector will have the same coordinates as its corresponding basis vector.

But by definition $(i)^2 = -1$. So to ensure $\langle \mathbf{e}^2, \mathbf{e}_2 \rangle = 1$, it must be that:

$$\mathbf{e}^2 = \begin{bmatrix} 0 & -i \end{bmatrix}$$

Intriguing! What 1 and i have in common is that they are unitary (having modulus 1). On the complex plane such numbers form the unit circle $e^{i\theta}$ (§2.5). When $\theta = \pi/4$ (45°) the real and imaginary ingredients are balanced, making an equal contribution to it having modulus 1:

$$\frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2}}$$

Both terms squared equal a half, so Pythagoras finds the "length" of the number to be 1. An equivalent way to find the modulus is to multiply the number by its complex conjugate, where the sign of the imaginary part is reversed. So if:

$$\mathbf{e}_2 = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2}} \end{bmatrix}$$

then it must be that:

$$\mathbf{e}^2 = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} - \frac{i}{\sqrt{2}} \end{bmatrix}$$

The covector coordinates are not the same: they are the conjugates of the vector coordinates. Of course, a real number having zero imaginary part, it is equal to

its own conjugate, which is why our original "same coordinates" rule worked. But we have now shown how to amend it to cover complex vector spaces in a backward compatible way.

In matrix terms, a covector acts on a vector by a row multiplying with a column. By selecting an orthonormal basis according to whatever inner product exists, the inner product expressed in that basis becomes the dot product, so a vector is able to "act on" any other vector just like it was a covector. A vector can act on itself with the dot product to find its own norm-squared. But we generally expect the norm of a vector to be a positive real number, even though the vector may have complex components.

So given two vectors $\mathbf{a}$ and $\mathbf{b}$ defined by column matrices a and b, where before the simple form of the inner product was defined as:

$$(\mathbf{a}, \mathbf{b}) = (a^\intercal)b = a_n b_n$$

it is now given by:

$$(\mathbf{a}, \mathbf{b}) = (a^\dagger)b = (a_n^*)b_n$$

where we use the dagger † symbol to mean *conjugate transpose*, where we both transpose the matrix and also take the complex conjugate of every element. This is also called the *Hermitian transpose*.

Care has been taken above to use parentheses to clarify that † and * are operators acting on the single object to their left, but these are usually omitted:

$$(\mathbf{a}, \mathbf{b}) = a^\dagger b = a_n^* b_n$$

This might be misconstrued as suggesting that † is a binary operator acting on the two objects on either side of it, like $+$, so be wary.

Although this definition ensures the norm is positive and real, it has the downside of destroying the symmetry of the inner product between two different vectors:

$$(\mathbf{a}, \mathbf{b}) \neq (\mathbf{b}, \mathbf{a})$$

All we can say is that:

$$(\mathbf{a}, \mathbf{b}) = [(\mathbf{b}, \mathbf{a})]^*$$

This is sometimes called conjugate symmetry. This does mean that when taking the inner product of two different complex vectors, it matters to which one

we apply the dagger. In physics the convention is to take the conjugate of the left vector, as shown above, and in pure mathematics they take the conjugate of the right vector - *c'est la vie*.

3.11.2 Hermitian operators

A similar adjustment must be made to the idea of a symmetric matrix (§3.8.1), with their useful properties. Rather than requiring the matrix to equal its own transpose, it must equal its own Hermitian transpose:

$$M^\dagger = M$$

That is, every element is the complex conjugate of its diagonally opposing element:

$$M_{ij} = (M_{ji})^*$$

Therefore elements on the diagonal are real (they aren't moved by the transposition and so must equal their own complex conjugates). We call this a Hermitian matrix,

Regardless of the basis, given an inner product, if:

$$(\hat{M}\mathbf{a}, \mathbf{b}) = (\mathbf{a}, \hat{M}\mathbf{b})$$

then $\hat{M}$ is a Hermitian operator, and in an orthogonal basis it is represented by a Hermitian matrix.

Several other important facts about Hermitian operators can be derived: their eigenvalues are all real and from their eigenvectors it is possible to choose an orthonormal basis that spans the space.

3.11.3 Complex unitary operators

We defined unitary operators in real spaces by saying that their inverse is their transpose. Unsurprisingly in complex spaces this definition has to be modified:

$$(O^\dagger)O = I$$

Also the intuitive understanding that a unitary operator performs a combination of rotation and reflection on a vector is no longer strictly true, because it may also "rotate" the individual coordinates in the complex plane (that is, multiply them by a unitary complex number).

Chapter 4

Quantum Mechanics

4.1 State Vectors and Dirac Notation

In quantum mechanics everything knowable about the state of some system is described in a vector, known as the state vector. The vector is from a vector space defined over the field of complex numbers, so it is important to use the correct definition of the inner product (§3.11) where we take the conjugate of one of the vectors, to ensure that the inner product of a vector with itself is a non-negative a real number.

The inner product in this context is written like this:

$$\langle \mathbf{a} | \mathbf{b} \rangle$$

Note that this is not the same as the notations we've used for a covector acting on a vector (§3.3), nor for the inner product (§3.6). The distinction between them is somewhat irrelevant here, as in the state vector space we will always be able to choose an orthonormal basis.

If the vectors $\mathbf{a}$ and $\mathbf{b}$ are represented by column matrices a and b (to spare ourselves, for the moment, from things we can't imagine, let's pretend we're discussing a finite-dimensional vector space), the above is equivalent to:

$$a^\dagger \, b$$

We can split this inner product notation into separate pieces, so we can write $\langle \mathbf{a} |$ to mean the vector whose matrix representation in some basis is a single row containing the complex conjugates of the elements in the single column of the matrix representing $| \mathbf{a} \rangle$.

It is valid to say that $\langle \mathbf{a}|$ is the covector of $|\mathbf{a}\rangle$, or that $\langle \mathbf{a}|$ as a function that extracts the coordinate of a basis vector $|\mathbf{a}\rangle$ from its argument, which will be some vector $|\mathbf{b}\rangle$, as in the expression $\langle \mathbf{a}|\mathbf{b}\rangle$. And in concrete matrix terms we can picture $\langle \mathbf{a}|$ as a 1-row matrix (a row vector) that is the dual of the 1-column matrix (column vector) $|\mathbf{a}\rangle$, and their corresponding coordinates are mutually complex conjugates.

And with this matrix representation in mind, it follows that we can write them the other way round from the inner product:

$$|\mathbf{b}\rangle\langle \mathbf{a}|$$

which must therefore produce a square matrix: the product of a column vector on the left and a row vector on the right. This is the *outer* product. A matrix can act as a vector-valued function of vectors: apply it to a vector to transform that vector to another vector. So:

$$|\mathbf{b}\rangle\langle \mathbf{a}|\mathbf{c}\rangle$$

See how the notation nicely suggests we bracket the $\langle \mathbf{a}|\mathbf{c}\rangle$ first as an inner product and thus a mere number. So we immediately know that the result will be the vector $|\mathbf{b}\rangle$ scaled by a number, i.e. it will be co-linear with $|\mathbf{b}\rangle$. We've measured $|\mathbf{c}\rangle$ against $|\mathbf{a}\rangle$ and used that to scale $|\mathbf{b}\rangle$.

Given an orthonormal basis $|\mathbf{b_n}\rangle$, we can picture it as a set of n column vectors, and expressed in their own basis they would be the standard basis. The outer product:

$$|\mathbf{b_n}\rangle\langle \mathbf{b_n}|$$

will produce a matrix with a single 1 in one place of the diagonal. So if we sum over all n, we get the identity matrix, a matrix that makes no difference to whatever vector it applies to. Spelling this out, if our vector space has just two basis vectors:

$$|\mathbf{0}\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and

$$|\mathbf{1}\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Then the outer product of $|\mathbf{0}\rangle$ with itself is just:

$$|\mathbf{0}\rangle\langle\mathbf{0}| = \begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix} = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}$$

and likewise of $|\mathbf{1}\rangle$ with itself:

$$|1\rangle\langle 1| = \begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}0 & 1\end{bmatrix} = \begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$$

And as predicted, summing those matrices gives the identity matrix:

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

Even if the $|\mathbf{b}_n\rangle$ were expressed in some other basis, the above summation would still be the identity matrix. Now for any vector $|\mathbf{c}\rangle$ we can construct for each n:

$$|\mathbf{b}_n\rangle\langle\mathbf{b}_n|c\rangle$$

The $|\mathbf{b}_n\rangle\langle\mathbf{b}_n|$ operator is called a projection operator, because it projects its argument onto the subspace spanned by $|\mathbf{b}_n\rangle$, resulting in a component vector of the argument in the direction of $|\mathbf{b}_n\rangle$. Clearly if we act with the same projection operator again on that result, nothing will change, because it's already projected. This is a way of defining a projection operator: it's idempotent.

And the sum of all those resulting vectors for all n will just be $|\mathbf{c}\rangle$, of course, because we've done the equivalent of acting with the identity operator.

4.2 Hilbert Spaces

The vector spaces used to represent physical states are examples of Hilbert spaces, which is a category that includes most of the familiar examples (e.g. a simple Euclidean vector space is also a Hilbert space). A Hilbert space is an inner product space with certain requirements, but they are very loose requirements, so it also includes more exotic situations than we encounter elsewhere in physics:

- scalars may be complex,
- despite which, there is an inner product that we can use to get a non-negative real number for the modulus of a vector: $\sqrt{\langle a|a\rangle}$, and
- the space may be infinite dimensional.

The latter possibility includes infinities that are continuous (uncountable). Such vectors cannot be represented by a column of discrete values, not even an infinitely long column. Instead we have to specify a complex-valued function over a continuous (real) variable. Such functions can be added and scaled, as is required of a vector, and so they qualify as elements of a vector space (§3.2) and we therefore have no choice but to admit that they are vectors.

The real parameter of such a function is analogous to the integer index that labels the rows in a column vector; instead of fetching the ith component by its position in the column, we evaluate the function with some real value x to get its component "at" x.

Similarly, whereas the inner product over discrete components is:

$$\langle \mathbf{a} | \mathbf{b} \rangle = \sum_i a_i^* b_i$$

the inner product over functions f and g of a real variable x is:

$$\langle \mathbf{f} | \mathbf{g} \rangle = \int_{-\infty}^{+\infty} f(x)^* g(x) dx$$

This is also called the overlap integral, because it measures the extent to which the two functions overlap, but it is most definitely also the inner product between two vectors. Thus we can in some sense find the square of the "length" of a function: $\langle \mathbf{f} | \mathbf{f} \rangle$. This sounds like gibberish, but it is an unavoidable consequence of the definition of a vector space, which is abstract enough to admit a space of possible functions.

4.3 Physical Interpretation

To interpret the state vector physically, we choose a basis so we can resolve it into components. Our choice of basis has to do with the observable quantity we are presently interested in, such as position, momentum, orientation or energy. If it may take on any real value, the state vector will have to be a function of that value; if it may only take on certain discrete values, it can be a column vector (albeit sometimes one with infinitely many rows) in which each row corresponds to one of those possible discrete values that the observable may exhibit when measured.

The information available from the state vector is, in general, probabilistic. Each component, being a complex number, is related to the probability of the observable quantity taking on the value represented by that component. The squared modulus of the component (its value multiplied by the complex conjugate of its value) is the probability of obtaining that value, or if the state vector is a function $f(x)$, then:

$$\int_a^b f(x)^* f(x)\, dx$$

is the probability that x will have a value somewhere between a and b.

As a probability is a number between 0 and 1, it must be the case that the sum of the squared modulus of all the components (or the above integral from $-\infty$ to $+\infty$) must be 1. This is the same as saying that $\langle S|S\rangle = 1$ for any physically realistic state vector. Or to put it another way, the magnitude of a state vector is not significant, only the direction (i.e. the relative values of the components in some basis). We will always fix the magnitude to be 1.

Unsurprisingly, if one of the components is 1 and all the others are zero, the vector represents certainty that the observable has the value represented by that component. But this also means that the state vector is equal to one of the basis vectors. Thus the basis vectors for an observable represent exact values that the observable may exhibit when measured.

Further, a measurement of the observable (or more precisely, any interaction producing subsequent behaviour that could be used to infer the value of the observable) causes the state vector to change to the basis vector of that observable corresponding to the measured value. This change is (at least in this theory) assumed to be instantaneous and to have no mechanism that we can deduce anything further about.

Thus after measuring an observable, subsequent measurements of the same observable will with certainty produce the same result.

(This is not quite true in the continuous cases when the state vector is actually a function of a real variable. We don't expect to ever find such a system precisely aligned with a single base state, but instead to have at least some small spread of probabilities.)

4.4 Switching Basis

Having constructed a column representation of a state vector in one basis, relating to one observable, we can switch to another. The operation for doing this will depend on both the "before" and "after" bases (§3.9). A state vector contains everything knowable about a system, including all we can know about any of its observable quantities. By re-expressing the same state vector as a different set of components in terms of the basis associated with a different observable, we recover the probability distribution for that observable.

The operation that switches basis will be a matrix if we're dealing with a finite-dimensional space, or something analogous to that if the space is infinite. As always when using a matrix to transform a vector's components we need to be clear on whether we want to get a different vector in the same basis or the

same vector in a different basis. In this case we're talking about the latter; a state vector represents something physically real, and we're just changing how we describe it. On the other hand the choice of basis is not entirely arbitrary because a basis relates to an observable quantity.

Any basis we change to must still be orthonormal. Therefore the transformation must preserve the inner product between any pairs of vectors, i.e. it must be unitary, and if it's represented by a matrix then the Hermitian conjugate serves as the inverse:

$$\hat{U}\hat{U}^\dagger = I$$

4.5 Operators Representing Observables

Observables have an associated operator, which maps vectors to different vectors. This has nothing to do with a change of basis. In QM when we talk about the observable's associated operator, we are talking about something that is not directly of any use for converting between bases (it is not unitary, for one thing), though it will indicate how we could perform such an operation.

An observable operator can be applied to a state vector as a kind of test, but it is much more powerful when we picture it applying to every possible state vector (that is, all unit vectors in the space) to find out how it affects them.

In QM operators associated with observables are Hermitian or self-adjoint, meaning that for an operator $\hat{O}$:

$$\langle \mathbf{a}|\hat{O}\mathbf{b}\rangle = \langle \hat{O}\mathbf{a}|\mathbf{b}\rangle$$

This has a few useful implications:

- in the discrete finite vector case, operators can be represented as a matrix O, $O^\dagger = O$, or $O_{ij} = O^*_{ji}$, so the main diagonal elements are real,
- regardless of representation, eigenvectors (§3.8) with distinct eigenvalues are orthogonal and complete (they span the space, so you can take a unit vector in each of these orthogonal directions and you have an orthonormal basis) and
- regardless of representation, their eigenvalues are real.

Think of the analogy of a Euclidean real plane vector space, and a symmetric 2×2 matrix M operating on it. The eigenvectors are lines in the plane along which vectors do not change direction, only magnitude, when the operator is applied. Because the matrix is symmetric ($M_{ij} = M_{ji}$) these lines are orthogonal. So it is with an Hermitian operator in a complex space, with only the

added complication of needing to be careful about taking the complex conjugate when comparing diagonally opposite elements.

The basis vectors of the observable are just unit vectors that are eigenvectors of the operator. That is, if you apply the observable's operator to every possible state vector, a subset of them will be scaled (by a potentially complex factor) without any change to their alignment. There will be a set of orthogonal unit vectors that pass this "alignment preserving" test, and these form the basis of the observable.

In other words, quantum mechanics is substantially about:

- defining the operator for an observable,
- solving the eigenvalue equation for that operator (that is, finding its eigenvectors and their associated eigenvalues)
- using the eigenvectors as a basis for representing state vectors,
- assuming that when the observable is measured, the state will snap into alignment with one of those eigenvectors,
- interpreting a coordinate in that basis as a complex amplitude whose mod-square is the probability that the state will align itself with that basis vector,
- interpreting the eigenvalue associated with the basis vector as the measured value of the observable (the eigenvalues of Hermitian operators are real numbers, fortunately.)

If a system's state vector matches one of these eigenvectors, then the system is already in an eigenstate and if the observable is measured, the result will with certainty be the eigenvalue associated with that eigenstate.

Otherwise, the state vector will be a linear combination of the eigenstates, and if the observable is measured and found to have a particular value, then the state vector will have instantaneously realigned itself with an eigenvector having that eigenvalue.

For the avoidance of confusion:

- Applying the operator to the state is not the same as "making a measurement"; that is something you do in the real world, where you may not even have any knowledge of the state prior to the measurement.
- Applying the operator to the state does not reveal the probability of anything (but see below about finding the expectation value.)
- Applying the operator to the state is not the same as changing its coordinate representation to be in the basis associated with the observable for that operator.

If you apply the operator to a state vector (which is, as always, of unit magnitude) and it is already aligned with an eigenstate, then its alignment will not be affected and its magnitude will change to experimental value you would get if you measured the observable in a system in that state. If it is not aligned with an eigenstate, the vector's alignment will change, but it will not snap into alignment with an eigenstate, which is what happens when a real world measurement is performed.

But if you solve the eigenvalue equation for the operator, you will know the complete basis of the observable, along with the eigenvalue associated with each basis vector, and you can then resolve our state vector against those basis vectors. Then you will have a set of coordinates that serve as probability amplitudes for the associated (measurable) eigenvalues.

In addition, there is a meaningful interpretation for the result of applying the operator for an observable to a given state vector: the inner product of the resulting vector with the original state vector gives the expectation value (§1.3) of the observable.

While we've discussed all this in terms of more easily pictured finite-dimensional vectors with discrete complex components, all the same concepts translate to complex-valued functions of an integer or real parameter.

4.6 Euclidean Visualisation

If we simplify radically, keeping to real numbers and two-dimensional Hilbert space, which is to say, the Euclidean plane, it becomes possible to draw pictures and invoke familiar geometric concepts. This is just a real-valued analogy for the complex-valued situations we will encounter in QM, so it has certain limitations, but it is accurate in many important details.

A state $|\psi\rangle$ is a vector. *It is not intrinsically in any basis.* But it is certainly of unit length.

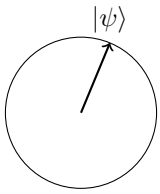


Figure 4.1: Anonymous state vector

Suppose the system can be found to be in one of two moods, $|a\rangle$ (affable) and $|b\rangle$ (bored). These will correspond to two orthonormal state vectors:

Naturally ψ can be resolved into a pair of coordinates by using the orthonormal vectors of mood as a basis. We are tempted to say that the mood is presently

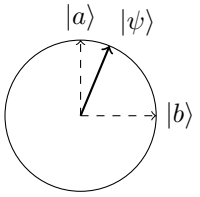


Figure 4.2: Mood basis

more closely aligned with affable rather than bored, but must always remember that the mood can only ever be measured to be exactly affable or exactly bored (upon which it will snap into alignment with $|a\rangle$ or $|b\rangle$ accordingly). It is just more likely to found affable, with probability given by the square of the coordinate given by the inner product $\langle a|\psi\rangle$.

Also the system can be found to be listening to music in one of two genres, $|c\rangle$ (country) or $|d\rangle$ (disco):

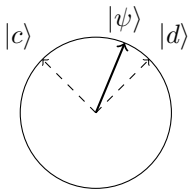


Figure 4.3: Genre basis

Against this genre basis the coordinates for the same $|\psi\rangle$ are clearly going to be different from what they were in the mood basis, and our $|\psi\rangle$ is leaning more toward disco than country. If you tilt your head to the left[1] so as to align $|d\rangle$ with the horizontal, pointing right, and $|c\rangle$ with the vertical, pointing up, then $|\psi\rangle$ will appear to be closer to horizontal than vertical.

Returning to the mood observable, there is an operator $\hat{M}$ associated with that observable. We can think of the operator as acting on all possible state vectors, represented by an evenly-spaced selection of them.

After the operator has done its work, the adjusted vectors fit into an ellipse rather than a circle: they are no longer all unit vectors. There are just two directions along which the vectors preserved their alignment: these are the eigenvectors of $\hat{M}$. They are orthogonal. This is how we discovered the basis vectors $|a\rangle$ and $|b\rangle$, by finding the states $|\psi\rangle$ for which:

$$\hat{M}|\psi\rangle = m|\psi\rangle$$

[1] That is, apply a unitary operator

Figure 4.4: Effect of operator $\hat{M}$

(a) Before (b) After

where m is just a number, i.e. the directions along which $\hat{M}$ does not change the alignment of the vector, only the length. Also the scaling factor along (say) the $|b\rangle$ direction is the numerical measurement that we interpret as the bored state.

Operators associated with observables, such as $\hat{M}$, are Hermitian, which (in this real vector space with only two orthogonal directions) means they can be represented by a symmetric 2×2 matrix, and will always have the effect of stretching or squashing along two orthogonal directions, thus picking out the two basis vectors for the observable.

And to be complete we should visualise the effect of the genre operator $\hat{G}$.

Figure 4.5: Effect of operator $\hat{G}$

(a) Before (b) After

It's another Hermitian operator, so it has again picked out two orthogonal directions along which it only applies a scaling.

There is no "true" basis against which a state vector is actually supposed to be measured. Basis vectors are just states that have a particular significance for certain operators.

The state could be aligned with $|a\rangle$, so the genre would be uncertain, and then it could become aligned with $|c\rangle$ and then the mood would be uncertain.

To relate all this back to more realistic QM scenarios:

- instead of restricting to real scalars, we allow complex scalars,

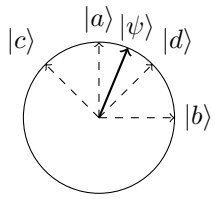


Figure 4.6: No special basis

- as well as just two orthogonal directions in state space, we allow infinitely many, even a continuum (as with position and momentum),
- we use the Hermitian inner product, taking the complex conjugate on the left, which means that the inner product of a vector with itself will always be real and positive, but the inner product between two different vectors may be complex,
- operators cannot generally be represented by matrices due to the continuous nature of some state spaces, but where they can, the matrix is Hermitian or self-adjoint, meaning that it is equal to its own conjugate transpose, which is the complex equivalent of a symmetric matrix,
- orthogonal eigenvectors of a Hermitian operator may in some cases have the same eigenvalue, and thus represent states that cannot be distinguished between by means of a measurement of the observable (picture our circle of state vectors growing or shrinking uniformly in all directions and thus remaining a circle, instead of being distorted into an ellipse),
- to get a probability from a complex coordinate, we take the modulus squared, to ensure it's a real number.

4.7 The Wave Function

One way to approach QM initially is to consider the position and momentum of an electron. These are continuous variables, so we will be working entirely with state vectors that are represented by functions of real variables, and operators that transform functions.

We model this situation as a continuous complex-valued function of position and time, $\Psi(x, y, z, t)$, very often abbreviated to Ψ. We will sometimes also consider functions only of space, ψ. (This upper/lowercase distinction is quite widespread but not universally observed.)

By considering only one spatial dimension we can picture the wave function at one instant as a line, somewhere along which the electron could be found. At each point x on the line there is an associated complex plane (visualised as

normal to the line), with an arrow lying in it, pointing out from the line. This is the complex value of Ψ at that position x and time t.

The complex plane should not be confused with vectors. Any given snapshot of $\Psi(x,t)$ at some instant t, given by a function $\psi(x)$, is itself an entire vector. The position x labels a single infinitesimal component of the vector, and every such component is a complex number, which we can therefore visualise as a complex plane with an arrow on it.

So for example we could picture the arrows as making a corkscrew shape, rotating around the line such that the angle depends linearly on x, but the modulus of the complex value (the length of the arrow) happens to be constant in this example. This is the notional wave function for a free electron (no forces acting it) with a precisely defined momentum and therefore no defined position, something never observed in reality.

More generally, the arrow length will also vary with x, t. The arrow length at x determines the likelihood that the electron will be found at x. More precisely, the modulus-squared of Ψ, which can be calculated with $\Psi^*\Psi$, is proportional to the probability density:

$$\rho(x) = \Psi^*\Psi \tag{4.1}$$

Given the electron is in some region A between x_1 and x_2, the integral:

$$\alpha = \int_{x_1}^{x_2} \Psi^*\Psi \, dx$$

is *proportional* to the probability of finding the electron in A.

Recall that the product of a complex number and its own complex conjugate is a real number, and here we are doing $\Psi(x)^*\Psi(x)$, using the single complex value at position x, so the result will be real. But the complex conjugate is not a general purpose magic way to get a real number from a product of any two complex numbers; $\Psi(x_1)^*\Psi(x_2)$ need not be real.

If we compute the same integral β for some larger surrounding region B, we can compute the conditional probability:

$$P(A|B) = \frac{\alpha}{\beta}$$

That is: the probability of finding the electron in A *given that* it is somewhere in B is given by the fraction α/β.

If Ψ is suitably behaved (square-integrable; roughly, it goes to zero at some distance and does not become infinite anywhere) then we can compute the integral over the whole of our one dimension of space:

$$\alpha = \int_{-\infty}^{+\infty} \Psi^* \Psi \, dx$$

We can then include a factor of $1/\sqrt{\alpha}$ within Ψ to "normalise" it, such that integrating the normalised $\Psi^*\Psi$ over some region will directly give us the absolute (unconditional) probability of finding the electron in that region.

Some interesting things to note at this early stage:

- For the simple first example of the free electron with definite momentum, normalisation is not possible because the integral over all of space does not converge on a finite value.
- A global change in the amplitude of the function (scaling the entire function by some complex constant) is not a physically significant change; there is a set of wave functions $a\Psi$ for any complex constant a, which all mean the same thing. What matters is how the amplitude varies from place to place (the same will turn out to be true for the complex phase).
- To normalise, we have to find the sum over all space of the mod-squared wave function. Interpreting the wave function as a vector, we're taking the inner product of the vector with itself, so we are in a sense finding the "length"-squared of the wave function as a vector. We then can then use this factor to scale it to be a unit vector, but preserving the relative shape of the wave (that is, preserving the "alignment" of the vector).

4.8 Schrödinger Equation

Any wave can be described as a sum of many simple component waves. (It is interesting that we use the word "component"; they are also basis vectors, so in vector terminology we should use the word component to refer to the complex constant factor applied to each simple wave included in the sum).

Each individual component wave has *two* parameters:

- if we nominate a fixed point in space, there is a frequency of oscillation, ν
- if we freeze time, we can measure the wavelength, λ, the distance between adjacent peaks in space

These can be independently adjusted (do not be confused by the familiar example of EM waves, where wavelength and frequency are coupled due to the constant speed of light!)

So the component wave can be described by the complex exponential:

$$\Psi(x,t) = \exp\left[2\pi i(\frac{x}{\lambda} - \nu t)\right]$$

Pick any fixed point in space, so x is constant, and ν determines the rate of oscillation. Pick a fixed instant in time, so t is constant, and λ determines the distance between peaks. With both in play, we have a corkscrew complex wave pattern that is moving.

Anything we figure out for this model wave can be taken to be true for any linear combination of many such waves, in the sense that we can imagine decomposing some messy wave into a set of components, each component characterised only by two numbers.

Planck inferred the relationship between frequency and energy:

$$\nu = \frac{E}{h}$$

And de Broglie likewise for momentum and wavelength:

$$\lambda = \frac{h}{p}$$

So we can write the wave function very neatly in terms of energy and momentum instead:

$$\Psi(x,t) = \exp\left[\frac{i(px - Et)}{\hbar}\right]$$

Nothing much has changed: as before, we have two parameters shaping a complex corkscrew wave. (We use $\hbar = h/2\pi$ for brevity because that combination isn't going away.) All that has changed is that we've got two parameters with a physical interpretation for something we've previously thought of as a "particle".

We can take the partial differential of the above w.r.t t or x, and the way that works with exponentials is strangely illuminating.

Doing t first:

$$\frac{\partial \Psi}{\partial t} = -\frac{iE}{\hbar}\exp\left[\frac{i(px - Et)}{\hbar}\right]$$

The constant factor is copied outside the exponential, which otherwise remains the same. So in fact:

$$\frac{\partial \Psi}{\partial t} = -\frac{iE}{\hbar}\Psi$$

We can tidy up by multiplying both sides by $i\hbar$:

$$i\hbar\frac{\partial\Psi}{\partial t} = E\Psi$$

The exact same procedure with x yields:

$$-i\hbar\frac{\partial\Psi}{\partial x} = p\Psi$$

But we can also take the second derivative and get:

$$-\hbar^2\frac{\partial^2\Psi}{\partial x^2} = p^2\Psi$$

Returning to our physical interpretation, a free particle has energy that is purely kinetic, related to its momentum by:

$$p^2 = 2mE$$

(This is just $\frac{1}{2}mv^2$ smushed into the definition of momentum, mv.)

Substituting the Planck and de Broglie relations:

$$\frac{\hbar}{2m} = \lambda^2\nu$$

In general a corkscrew wave is governed by two independent parameters:

- momentum, which goes with wavelength (and the x coordinate)
- energy, which goes with frequency (and the t coordinate)

We've now coupled them, making them no longer independent. But we've also added a new parameter: the particle's mass. For a free particle of a given mass, if you know the momentum you know the energy, and vice versa. Equivalently, if you know the wavelength you know the frequency, and vice versa.

Returning to the classical relationship between momentum, energy and mass, we can use it to rewrite our expression for $p^2\Psi$, substituting into the R.H.S. to easily obtain:

$$-\hbar^2\frac{\partial^2\Psi}{\partial x^2} = 2mE\Psi$$

And as we also have an expression for $E\Psi$, let's isolate that:

$$E\Psi = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2}$$

and insert our $E\Psi$ expression:

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2}$$

So, recalling that Ψ is an abbreviation for $\Psi(x,t)$, a complex valued function of space and time, now we have a differential equation that relates only these things:

- $\hbar$, Planck's constant, a universal fixed real number with units of joules-seconds, very accurately determined by experiment, not something we can adjust to fit this equation to different scenarios
- i, which just provides a 90°phase shift
- the first partial derivative of Ψ w.r.t. to time, which is another function of space and time that tells you how Ψ is changing
- m, the mass of the particle
- the second partial derivative of Ψ w.r.t. space.

This means that from a snapshot ψ (at a specific instant of time) of the wave function of a particle with a known mass, so you have its shape in space, you can find the second derivative of that shape w.r.t. space, then multiply that by $i\hbar/2m$ and you have the the first partial derivative of Ψ w.r.t. to time. That is, a snapshot contains complete information about the past and future of the wave; it tells you how to compute every past and future state.

So far, so kind-of rigorous. The situation becomes vaguer when we introduce a force field acting on the particle.

Schrödinger himself seems to have mostly taken a guess and found that the resulting equation agreed with several previously unexplained experimental results. Many widely used textbooks don't even give any background for it but merely state it. More advanced theory can be used to derive it, e.g. it is a low-energy approximation of QED.

The full classical account of the energy of a particle is:

$$E = \frac{p^2}{2m} + V$$

where the potential is a function $V(x)$. Realistically it will also be a function of t, but later we're going to pretend it isn't.

Some authors note that by multiplying the above throughout by Ψ:

$$E\Psi = \frac{p^2\Psi}{2m} + V\Psi$$

we obtain some scaffolding into which we can plug in our expressions for $E\Psi$ and $p^2\Psi$:

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} + V\Psi \tag{4.2}$$

And this is the same as the free particle equation with the added $V\Psi$ term, and is the complete Schrödinger equation which governs the time evolution of Ψ.

The extra term doesn't change the important property that if you have a snapshot $\psi(x)$ taken of $\Psi(x,t)$ at a specific initial instant of time, then you know all future states (glossing over what happens when there is any kind of interaction, including measurements).

This is sometimes contrasted with Newton's 2nd law relating acceleration to force, acceleration being the second order derivative of the position w.r.t time. Each time we integrate we need to conjure up a constant of integration, and we have to integrate acceleration twice to get the position. The two constants we need to add are the position and velocity. Thus a snapshot of the position of a particle is not generally enough to know what is happening to it.

But a snapshot $\psi(x)$ taken of $\Psi(x,t)$ at some time is not just one number, but a continuous function giving a (complex) number at each point x along the line, so it is generously endowed with information. If we decompose the snapshot into component waves, each one has its own wavelength.

And if we multiple Ψ by some constant (possibly complex) factor, the result is still a solution to the function. Such arbitrary constant scale factors make no difference to the physical meaning; what matters is how the function varies from location to location (and from time to time). This is what allows us to normalise the function (where possible) to ensure that it sums to 1 over all of space.

4.9 Time Evolution

We can say little here about wave functions unless they can be normalised, i.e. wave functions that tend to zero at infinity. Assuming this is the case, if we integrate the PDF over all of space:

$$\int_{-\infty}^{+\infty} \Psi^*\Psi\, dx$$

we expect the result to be constant (if normalised, it should always remain 1 as time passes), i.e.

$$\frac{d}{dt}\int_{-\infty}^{+\infty} \Psi^*\Psi\, dx = 0$$

Note that as we are integrating over x, outside the integral x is not a variable. We can move the differentiation w.r.t. t inside the integral, but only we change it to partial, because inside the integral x is a variable:

$$\int_{-\infty}^{+\infty} \frac{\partial}{\partial t} \Psi^* \Psi \, dx = 0$$

Focusing on the inside of the integral, by the product rule:

$$\frac{\partial}{\partial t} \Psi^* \Psi = \frac{\partial \Psi^*}{\partial t} \Psi + \frac{\partial \Psi}{\partial t} \Psi^*$$

Now, the Schrödinger equation gives us an expression for the partial time derivative of the wave function by slightly rearranging (4.2):

$$\frac{\partial \Psi}{\partial t} = \frac{i\hbar}{2m} \frac{\partial^2 \Psi}{\partial x^2} - \frac{iV}{\hbar} \Psi$$

From this we can get the same for the complex conjugate:

$$\frac{\partial \Psi^*}{\partial t} = -\frac{i\hbar}{2m} \frac{\partial^2 \Psi^*}{\partial x^2} + \frac{iV}{\hbar} \Psi^*$$

Plugging those into our expression:

$$\frac{\partial}{\partial t} \Psi^* \Psi = \left[-\frac{i\hbar}{2m} \frac{\partial^2 \Psi^*}{\partial x^2} + \frac{iV}{\hbar} \Psi^* \right] \Psi + \left[\frac{i\hbar}{2m} \frac{\partial^2 \Psi}{\partial x^2} - \frac{iV}{\hbar} \Psi \right] \Psi^*$$

Multiplying out:

$$\frac{\partial}{\partial t} \Psi^* \Psi = -\frac{i\hbar}{2m} \frac{\partial^2 \Psi^*}{\partial x^2} \Psi + \frac{iV}{\hbar} \Psi^* \Psi + \frac{i\hbar}{2m} \frac{\partial^2 \Psi}{\partial x^2} \Psi^* - \frac{iV}{\hbar} \Psi \Psi^*$$

The second and fourth terms cancel each other:

$$\frac{\partial}{\partial t} \Psi^* \Psi = -\frac{i\hbar}{2m} \frac{\partial^2 \Psi^*}{\partial x^2} \Psi + \frac{i\hbar}{2m} \frac{\partial^2 \Psi}{\partial x^2} \Psi^*$$

Also there's a common factor we can pull out:

$$\frac{\partial}{\partial t} \Psi^* \Psi = \frac{i\hbar}{2m} \left[\frac{\partial^2 \Psi}{\partial x^2} \Psi^* - \frac{\partial^2 \Psi^*}{\partial x^2} \Psi \right]$$

Recall that we are working out an expression for this because it appears inside an integral over all space:

$$\int_{-\infty}^{+\infty} \frac{i\hbar}{2m} \left[\frac{\partial^2 \Psi}{\partial x^2} \Psi^* - \frac{\partial^2 \Psi^*}{\partial x^2} \Psi \right] dx$$

Now the fundamental theorem of calculus is that integration is the inverse of differentiation, so there is clearly some redundancy here in that we are taking the second partial differential w.r.t. x only to then integrate over all x.

To make this explicit:

$$\frac{\partial}{\partial t} \Psi^* \Psi = \frac{i\hbar}{2m} \left[\frac{\partial}{\partial x} \left(\frac{\partial \Psi}{\partial x} \Psi^* - \frac{\partial \Psi^*}{\partial x} \Psi \right) \right] \tag{4.3}$$

The integral and the partial differentiation w.r.t. x cancel out to give us an expression that we can evaluate at the two limits and take the difference:

$$\frac{d}{dt} \int_{-\infty}^{+\infty} \Psi^* \Psi \, dx = \frac{i\hbar}{2m} \left[\frac{\partial \Psi}{\partial x} \Psi^* - \frac{\partial \Psi^*}{\partial x} \Psi \right] \Bigg|_{-\infty}^{+\infty}$$

If we do that, we will have an expression for the rate of change, w.r.t. to time, of the integral of $\Psi^* \Psi$ over all space.

But at these limits, we've said Ψ goes to zero, so as to be normalisable, making the whole expression zero at those limits. So in fact we've shown that, as we wanted:

$$\frac{d}{dt} \int_{-\infty}^{+\infty} \Psi^* \Psi \, dx = 0$$

So if it is possible to normalise a wave function at all, and it satisfies (4.2), then the constant of normalisation lives up to its name: it is the same for all time.

4.10 Motion

Given this abstract notion of an electron being entirely represented by a complex-valued function of position, how can we make sense of an electron moving?

Supposing the wave function is more concentrated in some region, it makes sense to compute the expectation value of the position variable:

$$\langle x \rangle = \int_{-\infty}^{+\infty} x \, \rho(x) \, dx$$

Substituting our definition of ρ from (4.1):

$$\langle x \rangle = \int_{-\infty}^{+\infty} x \, \Psi^* \Psi \, dx$$

remembering always that Ψ is short for $\Psi(x,t)$, so $\langle x \rangle$ is also a function of t, and so this gives us a way of thinking about motion: the way the expectation value of the position changes with time.

$$\frac{d}{dt}\langle x \rangle = \frac{d}{dt} \int_{-\infty}^{+\infty} x \, \Psi^* \Psi \, dx$$

We can rearrange to move the derivative inside the integral, giving:

$$\frac{d}{dt}\langle x \rangle = \int_{-\infty}^{+\infty} x \frac{\partial}{\partial t} \, \Psi^* \Psi \, dx$$

Like before, it's the t-derivative of something that depends on x, inside the integral over x we clarify that it is the partial derivative, and therefore x is a constant for that derivative.

And borrowing from (4.3) we can rewrite this as:

$$\frac{d}{dt}\langle x \rangle = \frac{i\hbar}{2m} \int_{-\infty}^{+\infty} x \frac{\partial}{\partial x} \left(\frac{\partial \Psi}{\partial x} \Psi^* - \frac{\partial \Psi^*}{\partial x} \Psi \right) dx$$

This isn't as simple as before where we cancelled out the integration and the differentiation, because of the pesky x. But the good news is this is the easiest ever opportunity for integration by parts. Recall:

$$\int u \frac{dv}{dx} dx = uv - \int v \frac{du}{dx} dx$$

So u is just x and to get v we have to calculate it at the limits:

$$v = \frac{\partial \Psi}{\partial x} \Psi^* - \frac{\partial \Psi^*}{\partial x} \Psi \Big|_{-\infty}^{+\infty}$$

Plugging them in:

$$x \left(\frac{\partial \Psi}{\partial x} \Psi^* - \frac{\partial \Psi^*}{\partial x} \Psi \right) \Big|_{-\infty}^{+\infty} - \int_{-\infty}^{+\infty} \left(\frac{\partial \Psi}{\partial x} \Psi^* - \frac{\partial \Psi^*}{\partial x} \Psi \right) \frac{dx}{dx} dx$$

As before, with Ψ vanishing at infinity the first term can be removed, and of course dx/dx is 1. Finally the above is just the integral from our $\langle x \rangle$ expression, so:

$$\frac{d}{dt}\langle x\rangle = -\frac{i\hbar}{2m}\int_{-\infty}^{+\infty}\left(\frac{\partial\Psi}{\partial x}\Psi^* - \frac{\partial\Psi^*}{\partial x}\Psi\right)dx$$

Having unwrapped one layer with integration by parts we can pull the same trick with $\frac{\partial\Psi^*}{\partial x}\Psi$, with $u = \Psi$ and $v = \Psi^*$, which once again means the uv term is zero, leaving:

$$-\int_{-\infty}^{+\infty}\frac{\partial\Psi}{\partial x}\Psi^*$$

So putting this back into $\langle x\rangle$:

$$\frac{d}{dt}\langle x\rangle = -\frac{i\hbar}{2m}\int_{-\infty}^{+\infty}\left(\frac{\partial\Psi}{\partial x}\Psi^* + \frac{\partial\Psi}{\partial x}\Psi^*\right)dx$$

The two identical terms cancel with the 2 on the bottom of the fraction, so:

$$\frac{d}{dt}\langle x\rangle = -\frac{i\hbar}{m}\int_{-\infty}^{+\infty}\frac{\partial\Psi}{\partial x}\Psi^* dx$$

If we think of the rate of change of $\langle x\rangle$ as the expectation value of the velocity, or $\langle v\rangle$, we can multiply by m to get $\langle p\rangle$, which actually cancels the m.

$$\langle p\rangle = -i\hbar\int_{-\infty}^{+\infty}\frac{\partial\Psi}{\partial x}\Psi^* dx$$

4.11 Operators Again

Another way to describe what we're doing here is rediscovering operators. To apply an operator $\hat{O}$ and get its expectation value $\langle O\rangle$, the recipe is:

$$\langle O\rangle = \int_{-\infty}^{+\infty}\Psi^*\hat{O}\Psi\, dx$$

How does this relate to our previous discussion about observable operators (§4.5)? We said that the operator for an observable is Hermitian, so it has orthogonal eigenvectors, and if the state vector is equal to an eigenvector then the observable, when measured, will be certain to equal the eigenvalue of that eigenvector. Our wave function at an instant in time $\psi(x)$ is a vector. To get a coordinate from that vector, we evaluate the function for some position x, and so the vector has a "coordinate" for every point in space. Therefore it is a vector expressed in the "position basis".

If the particle is very precisely localised, the function's value (the coordinates) will be zero everywhere except at that precise location. At the theoretical extreme, it will zero everywhere except at an infinitesimal single position (§1.4.2). That is, it will be a basis vector in the position basis.

An observable operator has to scale its eigenvectors by the value that would be measured for a state equal to that eigenvector. That is exactly what happens if we multiply $\psi(x)$ by x: if it is a pure spike (a complex value of modulus 1) at some position x_1, and zero everywhere else, the spike (and thus the whole vector) will be scaled by the value x_1. Whereas if it isn't a pure spike (not a position eigenvector), each non-zero value will be multiplied by a different value (its own position value), which will distort the shape of the function (or equivalently, change the "direction" and magnitude of the vector).

We also mentioned in passing that if we apply an observable's operator to a specific state vector, we get an adjusted vector, and if we take the inner product between the original state vector and the adjusted vector, the resulting scalar value will be the expectation value of the observable. So this is just another way of writing down the above integral:

$$\langle O \rangle = \langle \Psi | \hat{O} | \Psi \rangle$$

Because Ψ is a function of x and t, by integrating over all x we get a function of time, telling us the evolving expectation value of whatever observable the operator represents. To remove a little complexity we'll switch to considering an instant of time so $\psi(x)$ is all we need.

This "operator sandwich" pattern is intuitively sensible when we apply the position operator to a wave function of position, because this fits precisely with how we understand the expectation value to be computed: it is the sum of every possible value multiplied by its probability of occurring. $\hat{x}|\psi\rangle$ is just $x\psi(x)$. If we multiply that by $\psi(x)^*$ then it will be x multiplied by the probability of measuring the position to be x; clearly then the integral over all space will be the expectation value $\langle x \rangle$.

So in the position basis, the position operator $\hat{x}$ is just x itself:

$$\langle x \rangle = \int_{-\infty}^{+\infty} \psi(x)^* \hat{x} \psi(x)\, dx = \int_{-\infty}^{+\infty} \psi(x)^* x \psi(x)\, dx$$

The momentum operator $\hat{p}$, which we discovered above by looking for the expectation value of momentum, is $-ih\frac{\partial}{\partial x}$:

$$\begin{aligned}\langle p \rangle &= \int_{-\infty}^{+\infty} \psi(x)^* \hat{p} \psi(x)\, dx \\ &= \int_{-\infty}^{+\infty} \psi(x)^* (-ih\frac{\partial}{\partial x}) \psi(x)\, dx \\ &= -ih \int_{-\infty}^{+\infty} \psi(x)^* \frac{\partial \psi(x)}{\partial x}\, dx\end{aligned} \tag{4.4}$$

Compared to the position operator, it is somewhat less obvious what the momentum operator is doing to produce an expectation value for momentum. The intuitive process would be to sum (over all possible momenta) the product of each momentum and its probability of being measured. That would seem to require a wave function of momentum instead of position, and an integral over all momenta. And yet here we still have an integral over all positions, involving a wave function of position. *We have not yet changed basis.*

To represent a vector as a set of coordinates in some basis, we extract each coordinate of the vector by performing the inner product between the unit basis vector for that coordinate and the vector in question. But for these functions of real parameters like position and momentum, the definition of a vector (that is, the set of its coordinates) is given by a function of the real parameter, the values of that function being the coordinates, each coordinate labelled with the real parameter, and the inner product is an integral over all values of the real parameter.

Momentum basis vectors in the position basis are of the form:

$$\phi(x) = \frac{1}{\sqrt{2\pi\hbar}} e^{ipx/\hbar}$$

That is, they are a function of position (we're still in the position basis, so the "components" of the vector are labelled by positions), but there is a momentum variable p in the definition. That value of p determines which basis vector of momentum this $\phi(x)$ describes.

The inner product of this with our state vector $\psi(x)$ (which is also expressed in the position basis) is the integral:

$$\int_{-\infty}^{+\infty} \psi(x)^* \frac{1}{\sqrt{2\pi\hbar}} e^{ipx/\hbar}\, dx$$

This yields a single complex number that is the "coordinate" associated with momentum p of our state vector in the momentum basis. So the complete description of our state in the momentum basis is a function of momentum:

$$\psi(p) = \int_{-\infty}^{+\infty} \psi(x)^* \frac{1}{\sqrt{2\pi\hbar}} e^{ipx/\hbar}\, dx$$

Multiplying by a complex exponential inside an integral in this way is recognisable as the Fourier transform. But also, now we have $\psi(p)$ we can get the expectation value of momentum in exactly the same way as we did for position: by integrating (over all momenta) $\psi^* \hat{p} \psi(p)$:

$$\langle p \rangle = \int_{-\infty}^{+\infty} \psi(p)^* \hat{p} \psi(p)\, dp$$

In the momentum basis, the $\hat{p}$ operator is just multiplying by p (exactly like the position operator in the position basis.)

$$\langle p \rangle = \int_{-\infty}^{+\infty} \psi(p)^* p \psi(p)\, dp$$

And $\psi(p)^* \psi(p)$ is the mod-square of the coordinate for p, that is, the probability that the momentum has the value p.

The point is if we do change basis, we are not materially changing what is calculated. It is perhaps easier to comprehend this intuitively through the analogy with regular vectors, and by remembering that vectors are basis independent objects. The observable operator acts on the state vector, in general changing both its length and alignment (unless the state vector happens to already be an eigenstate of the observable, in which case only the length changes). But this action would be the same regardless of the basis we are working in, and so it isn't necessary to get hung up on that point. Likewise, projecting the operated-on vector back on to the original state vector, to produce a scalar expectation value, is a geometrical, basis-independent operation. The inner product depends on the relative lengths and the angle between the two vectors. As long as when we change basis we do so in a way that preserves the inner product (that is, by a *unitary* operator), then the choice of basis is physically irrelevant.

Therefore, in this recipe for the expectation value, the operator $\hat{O}$, the state vector $|\psi\rangle$ and the adjusted vector $\hat{O}|\psi\rangle$ should all be understood as having an independent existence from any choice of basis:

$$\langle \psi | \hat{O} | \psi \rangle$$

That expression is a scalar value, and is the same regardless of the basis we work in (if we switch basis, we must use a unitary operator so the inner product between any two vectors is unaffected by the change of basis). When we want to calculate the value, we use the same basis throughout.

4.12 Time Independent Potentials

In the Schrödinger equation, if the potential V is constant everywhere (and thus may as well be zero everywhere), it reduces to the free particle equation that fell out automatically from the fact that kinetic energy is tied to momentum. A real particle wave function will be some shape that is smooth, differentiable and vanishes at infinity, but it nevertheless always be thought of as the sum of an infinite set of contributing simple corkscrew waves, which means its behaviour is entirely predictable from a time-independent snapshot of the wave $\psi(x)$.

If the potential is a more interesting function it gets trickier. To understand the effect of varying t and x separately, we can suppose the existence of two functions $\psi(x)$ and $\phi(t)$ that when multiplied give us $\Psi(x,t)$.

It is not generally true that this is possible. Even something as simple as $\Psi(x,t) = x + t$ can't separated into a product of two functions of x and t. It's obviously true that solutions to the zero-potential Schrödinger equation can be separated, simply because we obtained it from the assumption:

$$\Psi(x,t) = \exp\left[\frac{i(px - Et)}{\hbar}\right]$$

which can easily be written as the product of two separate functions of x and t:

$$= \exp\left[\frac{ipx}{\hbar}\right] \exp\left[\frac{-iEt}{\hbar}\right]$$

But when a potential is included, and it is some complex function of x and t, that is no longer necessarily possible. But we can still discover some useful things by assuming the potential does not depend on t, and that is an acceptable approximation. So the whole time-dependent wave function is a product of two parts:

$$\Psi(x,t) = \psi(x)\phi(t)$$

Taking partials becomes ordinary differentiation, because the other factor is constant:

$$\frac{\partial \Psi}{\partial t} = \psi \frac{d\phi}{dt}, \frac{\partial^2 \Psi}{\partial x^2} = \frac{d^2\psi}{dx^2}\phi$$

So we just plug those into (4.2):

$$i\hbar\psi\frac{d\phi}{dt} = -\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2}\phi + V\psi\phi$$

Dividing by $\psi\phi$:

$$i\hbar\frac{1}{\phi}\frac{d\phi}{dt} = -\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2}\frac{1}{\psi} + V$$

To make this explicit, let's put the parameters on each function:

$$i\hbar\frac{1}{\phi(t)}\frac{d\phi(t)}{dt} = -\frac{\hbar^2}{2m}\frac{d^2\psi(x)}{dx^2}\frac{1}{\psi(x)} + V(x)$$

The LHS only depends on t, the RHS only depends on x (this wouldn't have worked without the simplifying assumption that V is independent of t). This means if we hold x constant, and therefore the RHS constant, this equation still holds even if we vary t! And of course vice versa. Which means both sides are equal to the same constant, which we will call E for a good reason (spoilers!)

Equating the LHS with E:

$$i\hbar\frac{1}{\phi}\frac{d\phi}{dt} = E \therefore \frac{d\phi}{dt} = -\frac{Ei}{\hbar}\phi \therefore \phi = e^{-iEt/\hbar}$$

The RHS isn't so neat, but:

$$-\frac{\hbar^2}{2m}\frac{1}{\psi}\frac{d^2\psi}{dx^2} + V = E \therefore -\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2} + V\psi = E\psi$$

Solutions for ψ will depend on V of course. But the whole wave function is therefore:

$$\Psi(x,t) = \psi(x)e^{-iEt/\hbar}$$

Why is this interesting? Because the more complicated space-sensitive part is frozen w.r.t. time, we can understand the time evolution by just looking at the extremely simple factor:

$$e^{-iEt/\hbar}$$

Whatever the solution to ψ, the complex value of every point in space is only changing by the above factor as time passes.

And that factor is really just $e^{i\theta}$ with the angle being $-Et/\hbar$, so we know the modulus of the value isn't changing; it's just going "round and round" clockwise in the complex plane.

And if the modulus isn't changing, the probability density isn't changing, so the particle isn't moving. Hence solutions of this type are known as *stationary*

states. The expectation value of the position is fixed, and so all other observables' expectation values are also constant, including energy.

The shape of a stationary state, as defined by the x-dependent factor, is a fixed structure which never deforms, but just tumbles over and over as time passes, like it's rotating on a spit, although the rate of rotation has $\hbar$ on the bottom of the fraction, which is about $10^{-}34$, so it's rotating extremely quickly.

4.13 Bound States

When we constructed the Schrödinger equation (4.2) we did so by building an expression for the total energy in terms of the particle's kinetic energy and a potential:

$$-\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} + V\Psi$$

Comparing this to our RHS differential equation:

$$-\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2} + V\psi = E\psi$$

So in one of these stationary states, ψ substitutes for Ψ, which is quite valid because the energy is constant, and so it's the same expression. We can extract an operator for the energy (the Hamiltonian):

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V$$

And in a stationary state, it's just:

$$\hat{H}\psi = E\psi$$

This is an eigenvalue equation: for some solutions ψ, the $\bar{H}$ operator has the same effect as multiplying by a constant, the energy eigenvalue E for that ψ. It can be shown that it's a Hermitian operator, so the eigenvectors for stationary states with distinct energy eigenvalues can be used to define an orthonormal basis.

Note that we're not talking about a single general operator for all situations. The definition of $V(x)$ will depend on the situation, and that will affect the solutions of the above eigenvalue equation.

There are several standard potentials that can be analysed in this way. The simplest is the infinite square well, a potential that is zero in a small region of width a but infinite outside that region, forcing the wave function to be zero

at the boundaries of the region and causing it to have eigenfunctions that are standing waves:

$$\psi_n(x) = \sqrt{\frac{2}{a}} sin(\frac{n\pi}{a}x)$$

with corresponding energy eigenvalues:

$$E_n = \frac{n^2\pi^2\hbar^2}{2ma^2}$$

The interesting thing about this is that the set of eigenfunctions, while infinite, is a countable infinity: there is a first state, a second state and so on. Despite this, by choosing an infinite subset of these states and summing them, we can generate any function. Strictly speaking, we can generate any periodic function that is zero at the boundaries, but the distinction between periodic and non-periodic functions is irrelevant if we have defined the value to be zero everywhere outside those boundaries; the shape of one cycle of a periodic function can be anything we like.

So there is no limit on what the initial state can be, but by decomposing that $t = 0$ state $\Psi(x, 0) = \psi(x)$ into a sum of these discrete waves, we know exactly how it will evolve over time. Each component eigenfunction evolves due to the "round and round" factor $e^{-iEt/\hbar}$. Because we're adding complex values at each point in space, even though those component values each have a time-independent modulus, the sum of them does not (think of a clock hand with another clock hand on the end of it, running at different rates). So this is a way to make non-stationary solutions. Wave packets that "move" can be composed by summing stationary states that do not. A particle confined inside a potential can be in a stationary state, or it can "slosh" from side to side in a complicated way due to being a linear combination of many such states.

So we have an exact solution for how any initial state will evolve inside an infinite well potential, assuming the total energy is constant. The energy, by definition, is the quantity that is preserved as time passes, and that's why energy eigenfunctions are of primary importance in analysing the time evolution of a system. To get an exact expression for the wave function as a function of time, we resolve it into a sum of energy eigenfunctions, each of which has its time evolution entirely determined by the factor $e^{-iEt/\hbar}$.

The energy eigenfunctions serve as a set of basis vectors. Why? Because the energy operator is Hermitian, so among its eigenvectors, any two that correspond to distinct eigenvalues will be orthogonal. For stationary states m and n:

$$\int \psi_m^* \psi_n dx = \delta_{nm}$$

This does not mean that $\psi_m^* \psi_n$ is zero everywhere if $m \neq n$, but it does mean that for every non-zero value pointing in some direction in the complex plane, there's another value of the same modulus pointing the opposite way, to balance it out.

So each possible energy value is represented by an eigenvector. We can create a weighted sum of them to make any possible state. We can use those weightings as the components of a vector describing a state. That is, the state of the lowest energy level is a column vector of numbers where the first component is 1 (or some complex number of modulus 1) and all the other components are 0.

There are two complications to this story:

- In some situations we find that multiple eigenvectors have the same eigenvalue, so we cannot infer the state vector from an energy measurement, a predicament known as *energy degeneracy*. We never see this for energy in one dimension of physical space, but in two or three it is unavoidable. Motion back and forth along the x-axis is physically different from such motion along the y-axis, but may be so similar as to require the same energy.
- We can't measure the energy with absolute precision anyway. No system ever really collapses so its state vector is known to be precisely one of the eigenvectors of whatever observable is being measured. It always remains a superposition to some extent.

States where the particle is trapped in a potential are known as *bound states*. In the absence of a potential, particles are in *scattering states*, where the possible energies are not quantised but continuous, so we have to use an integral to find the inner product:

$$\langle \alpha | \beta \rangle = \int \alpha^* \beta \, dx$$

If we knew a particle's exact location, α, our wave function of space $\psi(x)$ would have a single spike where $x = \alpha$ and be zero everywhere else. Alternatively if knew its exact momentum (and $p = h/\lambda$) our wave function would be a wave with a single wavelength. So we're dealing with Fourier transforms. At these extremes of certainty/uncertainty, one domain has a simple wave of infinite extent, and the other domain has a spike representing that wave. It works either way round.

Thus far we've been working in "position space", using functions of x, but alternatively we could work in "momentum space", where the functions are $\phi(p)$. If we knew a particle's exact momentum, $\phi(p)$ would be a spike, whereas if we knew its exact position, $\phi(p)$ would be a single-component wave.

4.14 Energy Degeneracy

Given two eigenvectors of an Hermitian operator, if they have different eigenvalues then they are orthogonal (intuitively they can't be colinear because a linear operator can't apply a different scaling to vectors that are colinear). But what about the converse? If they are orthogonal do they necessarily have different eigenvalues?

Given two eigenstates of the energy operator, ψ_1 and ψ_2, when we apply the operator it just scales them each by their own eigenvalue:

$$\hat{E}\psi_1 = E_1\psi_1$$
$$\hat{E}\psi_2 = E_2\psi_2$$

If $E_1 \neq E_2$, the two states must be orthogonal. What if we sum the two states and apply the energy operator to the result? The operator is linear, so applying it to a sum of states is the same as applying it to the states separately and then summing the result:

$$\hat{E}(\psi_1 + \psi_2) = \hat{E}\psi_1 + \hat{E}\psi_2 = E_1\psi_1 + E_2\psi_2$$

So the result is a linear combination of the two states. If $E_1 \neq E_2$, this result is not some single constant E_3 multiplied by the sum of the states:

$$\hat{E}(\psi_1 + \psi_2) \neq E_3(\psi_1 + \psi_2)$$

But if $E_1 = E_2$, then we can pull out a single constant (let's use E_1):

$$\hat{E}(\psi_1 + \psi_2) = E_1(\psi_1 + \psi_2)$$

In other words, any two eigenstates with the same eigenvalue can be added to get a third eigenstate. Obviously this is also true for any linear combination using weightings a, b, simply because if ψ is an eigenstate then so is $a\psi$:

$$\hat{E}(a\psi_1 + b\psi_2) = \hat{E}a\psi_1 + \hat{E}b\psi_2 = E_1a\psi_1 + E_2b\psi_2 = E_1(a\psi_1 + b\psi_2)$$

So this is a subspace of the whole vector space, a set of eigenvectors with the same eigenvalue, known (inevitably) as an eigenspace. Any scalar multiple of an eigenvector is also an eigenvector, so the line along which an eigenvector lies is (though not often) called an eigendirection (*eigenalignment* would be a better term), and is the simplest eigenspace that can arise with a linear operator.

In the simple case of a symmetric matrix operating on the plane, it will pick out two orthogonal directions along which it will perform a pure scaling. It may

stretch along one direction but squeeze the other. Any vectors colinear with these two directions will only be scaled, their directions unaffected, and thus are eigenvectors. All other vectors will be rotated, and so are not eigenvectors. The set of eigenvectors aligned with one of the directions will all have the same eigenvalue, and so they form an eigenspace.

In QM the magnitude of a state vector is not significant; we always normalise it to a unit vector anyway. Therefore it is truer to say that observable states are represented by eigendirections, and in a given direction we nominate the unit vector to be *the* eigenstate in that direction.

But in the higher dimensional complex vector spaces of QM, some operators will have eigenspaces that are not mere lines, but are themselves multidimensional. That is, there will be a set of vectors that are eigenvectors sharing the same eigenvalue, among which we can find an orthonormal basis of 2 or more dimensions.

The operation of scaling equally in all directions is a trivial example where all vectors are eigenvectors, and all have the same eigenvalue (the scaling factor), so they are all in the same eigenspace, and we can select any N that are not colinear to use as a basis of the N-eigenspace. Suppose we're dealing with a three dimensional vector space, and we scale up by a factor of 1.5 along the x and y directions, but shrink by 0.5 along the z direction. A vector aligned with either the x or y direction will be an eigenvector under this scaling operation, with the eigenvalue 1.5. So this is degeneracy: orthogonal eigenvectors with the same eigenvalues, sharing a planar eigenspace that is a proper subset of the whole space.

So while we can say confidently that if two eigenvectors of an Hermitian operator have different eigenvalues then they are orthogonal, we cannot claim the converse: two orthogonal eigenvectors do not necessarily have different eigenvalues. In some circumstances they do, but not all. In particular, by classical intuition, the system of a weight on a string swinging back and forth has a total energy related to the weight's maximum displacement, but it can swing along any axis, so there is an infinite set of states with the same energy. So it is in QM. The inability in some situations to determine the state from the energy is known as degeneracy.

4.15 Spin

Up to this point we've been mostly considering a specific application of the general model of QM, where the classical concepts of position and momentum, which are the building blocks for anything else we might measure, are represented as functions of a continuous variable. The function's value is a complex number that can be mod-squared to get the probability density of that variable. There is something intuitive about this in the space domain, which is why we start with that: it makes us think of an electron as being spread out through

space, and having a density that varies. Only when some interaction occurs does it appear to be concentrated entirely at one point. But the momentum of the electron is modelled the same way, and this is not by itself very intuitive. It only becomes a little clearer when we realise that the space representation is a wave, and we can model waves as a sum of simple component waves with a set of frequencies. The momentum function is the Fourier transform of the position function.

But now we're going to consider intrinsic angular momentum. This is introduced to explain how electrons are deflected in a magnetic field, behaviour which is classically suggestive of the electron spinning around an axis, although that physical interpretation seems impossible because the electron's radius (if it is non-zero) is extremely small. Antipodal points on the surface of the electron would have to be moving relative to each other faster than the speed of light. Pauli suggested glossing over this question and just accepting that electrons have an intrinsic angular momentum that cannot be interpreted in some comfortable classical way.

Chapter 5

Relativity

Newtonian mechanics is based on the notion that the passage of time is universal, and objects have motions that determine how their positions change with the passage of time.

Einstein (and Minkowski) overturned this. Space and time are dimensions of a combined *spacetime*. The orientations of the space and time axes are a matter of perspective.

Rather than a point particle in space that is in motion, picture a path through spacetime, made up of points called *events*. The standard term for this is a *world line*, but I'm going to call it an *event path* (as it is not necessarily a line).

So in such a structure there is no motion at all; it is fixed and permanent.

Straight segments of an event path correspond to uniform (non-accelerated) motion. What we call acceleration is any curved portion of the path.

To assert that a particle is "at rest" during some straight segment of its event path is to choose to align the time axis with that segment. To assert that a particle is "in uniform motion" is to choose a time axis that is not aligned with the particle's path.

Any diagram of spacetime we draw, with a time axis and space axis, necessarily requires us to choose a specific alignment for the time axis, and thus a space axis that is (from the perspective of one at rest) orthogonal to it.

5.1 Clock Arrays, Spacetime Grids

Consider particles that only have one degree of freedom, i.e. they take positions along a line. Nominate an origin on the line. At the origin, place an emitter of a pulse of light, and on either side of it, stretching off to infinity, place

probes such that they are spaced one light-second away from their immediate neighbours.

Each probe contains a digital clock that measures elapsed seconds, but which is initially paused so its value does not advance as time passes. Each probe's paused clock displays an elapsed time that is equal to the probe's distance from the origin in light-seconds (note: we intentionally say the distance, which is always positive, not the displacement, which would be negative on the left and positive on the right).

At the origin, our emitter has a paused clock showing zero. It simultaneously emits the pulse of light and starts its own clock. When each probe detects the light pulse arriving, it starts its own clock.

In this way, we create a line of evenly spaced clocks that are synchronised with the origin's clock. We could now (if we wanted to) discard any notion of the distinct identity of each clock and treat them as an array of indistinguishable synchronised clocks. But instead we will label each clock with its displacement from the origin, so clocks on the left of the origin have increasingly negative labels, and those on the right have increasingly positive labels.

A spacetime diagram of this construction would be an orthogonal grid.

- The intersections of the grid represent events where a specific clock ticked forward to a new whole number of seconds
- Each vertical line connects all such events for a single clock, so a vertical line *is* a clock, in that it is the event path of a clock.
- Each horizontal line connects the events where a clock ticks forward to show a higher number of whole seconds have elapsed.

On such a diagram we can imagine our perception of time passing as being represented by a horizontal "line of simultaneity" that sweeps up through the grid.

5.2 Curved paths

A rogue particle is now introduced. It is free to move left or right on the line. It can accelerate freely. It is depicted on our spacetime grid by a curved path. On its journey it visits several of our clocks, which are able to sense when it passes very nearby and make a note of the time (according to that clock) when such a close encounter occurs. Thus we can build up a record of the movement of the particle, consisting of pairs of position (the clock's label) and time (the clock's time) captured at each clock it passes.

Because the rogue particle is outnumbered, it appears very clear that the the rogue is in non-uniform motion against the background of our original array of probes.

To even the score, suppose that rather than one such rogue particle, we have an array of them, spaced out along the line, and each having its own clock. These rogue clocks are synchronised by a light pulse just as before.

All the particles in the rogue array have been programmed to carry out a predetermined sequence of accelerations, by firing little rocket thrusters. They all perform these accelerations in perfect unison and so remain the same distance from each of their neighbours. They are another co-moving array.

The result is that relative to each other, the rogue clocks are seemingly at rest, and could presume that it is our original array of clocks that are doing all the accelerating, and it being mere chance that their own rocket firings coincide perfectly with accelerations undertaken by other particles.

But nature is not fooled. At some basic level, the act of accelerating by firing a rocket is accounted for and is an objective fact, not something that can be defined away by a change of perspective. The straight line segments of an event path are objectively straight, and the curved segments are objectively curved. The only valid conversions between points of view must be linear, in that they never convert a straight path into a curved path or vice versa.

So we have two fundamentally different arrays of co-moving particles, one in uniform motion and the other in accelerated motion.

Even so, we can imagine following the path of one rogue particle across spacetime as its own clock ticks. Naively we might imagine rotating the diagram against our coordinate system as we trace the path, so as to keep the tangent to the path aligned with the time axis. After all, rotations are linear transformations that conserve the distance between each point and the centre of rotation. Each point moves in a circle around that centre.

Usually you'd assume that the invariant separation between points is captured by the Pythagorean formula:

$$r^2 = t^2 + d^2$$

But the correct form of rotation to use between a space and a time coordinate is hyperbolic rotation, such that each point moves on a hyperbola as we rotate our perspective, and the equivalent of a distance between events being conserved is called the *interval*, s:

$$s^2 = t^2 - d^2$$

It transpires that the interval between events is an objectively real fact that all observers agree on. The absolute structure of the universe is *spacetime*, which is the set of all events, and all observers agree on the interval between any two events.

5.3 Spacetime Metric

We can define a metric, a way of taking the inner product of two vectors, and to find the squared length of a vector we take the inner product of the vector with itself. The metric for Euclidean geometry is the Kronecker delta, δ, e.g. for three space dimensions:

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

According to that metric, the squared length of vector $\mathbf{v}$ in terms of its coordinates v^i is:

$$|\mathbf{v}|^2 = \sum_i \sum_j \delta_{ij} v_i v_j$$

As δ picks out the terms where $i = j$:

$$|\mathbf{v}|^2 = (v_1)^2 + (v_2)^2 + (v_3)^2$$

This is the familiar theorem of Pythagorus.

But to find the interval between two events in spacetime, we need to use a different metric, η (eta). Now our indices i, j can take four values, traditionally given as $0, 1, 2, 3$, with time being 0. So with a time dimension in addition to the three space dimensions, the correct metric happens to be:

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

where c is the speed of light, a quantity with cosmic significance.[1] In honour of this we can take as our unit of distance the light-second, so that c is 1.

Furthermore, as our particles only move along a straight line we only need one space dimension, so altogether the metric can be written as:

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

[1] Note that the signature of the diagonal of η could just as well be $(-, +, +, +)$ instead of $(+, -, -, -)$, and as with anything that makes no difference, debate has raged on for over a century.

So we have arrive at the modified form of Pythagorus such that the interval, s between two events separated by a distance d and a time t is given by:

$$s^2 = t^2 - d^2$$

It would be quite misleading to continue to picture this as a right triangle with d and t as the orthogonal short sides and s as the hypotenuse, because s has to remain constant as d and t both increase, which is impossible if the d and t sides remain orthogonal.

5.4 Interval Related To Proper time

A clock that travels inertially between two events separated by interval s will measure the elapsed proper time τ, but this is in fact identical to the interval:

$$\tau = s = \sqrt{t^2 - d^2}$$

This is obvious given that in its own coordinate system, it remains at coordinate zero, and so $d = 0$ and hence:

$$\tau = s = t$$

This underscores the invariant, universal, unambiguous nature of the interval between two events. Two synchronised clocks departing from some event, and arriving at another via different "routes" through spacetime, may each show a different elapsed proper time, and each's proper time is a measure of the "length" of the path it took (the sum of all the infinitesimal segments of interval along that path.)

But due to the nature of the metric, with time and space coordinates making opposite-signed contributions to the sum, less straight paths require *less* time to elapse for the particle taking them. Thus the straight line path between the events (that which would be taken by a uniform motion) is the *slowest* path possible, in that a particle taking that path will see its own clock advance by a greater duration than that of any particle taking some otherwise curved path.

5.5 Lorentz Factor

Consider a clock-carrying particle P that is in uniform motion relative to an array of co-moving clocks, A_n. Two events are of significance:

- P passes by A_0 when both P's and A_0's clocks read t_0
- P passes by A_1 when A_1's clock reads t_1.

As measured in the A frame, the elapsed time t between the two events is:

$$t = t_1 - t_0$$

Also in A the distance between the two events is the distance between the clocks:

$$d = A_1 - A_0$$

And with these two values, A can compute the constant velocity of P during its journey:

$$v = d/t$$

But things look a little different from the perspective of P. A uniformly moving clock regards itself as being at rest, so the interval between the two events is accounted for entirely by the passing of time, and so the elapsed time according to P is equal to the interval of the path it takes.

We know that the interval is an invariant, objective fact about reality, not something that changes based on perspective, and we also know that it equals the time measured by a clock moving inertially between events. So using the values for d and t obtained in A we can correctly calculate the elapsed time measured by P:

$$\tau^2 = t^2 - d^2$$

Pulling out a factor of t^2:

$$\tau^2 = t^2(1 - \frac{d^2}{t^2})$$

We can abbreviate this by using A's value for the velocity $v = d/t$:

$$\tau^2 = t^2(1 - v^2)$$

And unsquaring both sides:

$$\tau = t\sqrt{1 - v^2}$$

So the ratio that will convert the proper time τ measured by P back to the time coordinate separation t as measured in A is:

$$\gamma = \frac{1}{\sqrt{1 - v^2}}$$

This γ is the Lorentz factor. Note that as v is based on a time value t in seconds and a distance value d in light-seconds, it is a fraction of the speed of light.

If it had the value 1, that would be light speed. But at that precise value, γ is undefined, due to the zero on the bottom of the fraction. The elapsed time for any journey taken by a photon is always zero, and so there is no way to relate times recorded by a photon to times recorded by an array of clocks (a photon is not a clock).

Furthermore, at velocities greater than 1 (faster than light), the value is defined but unfortunately is imaginary, due the square root of a negative value. The square of the interval between events travelled between by faster-than-light particles is negative, and so the interval, which is the elapsed time recorded by a clock carried by such a particle, is positive-imaginary.

5.6 Lorentz Transformations

Two observers ρ and ϕ moving apart at speed v will label the same event with different spacetime coordinates: (t_ρ, x_ρ) and (t_ϕ, x_ϕ).

To transform back and forth between these systems we need a pair of matrices, each the inverse of the other, such that these invariants are conserved:

- the interval between events
- the speed of light

Note that for any

$$\begin{bmatrix} \gamma & v\gamma \\ v\gamma & \gamma \end{bmatrix}^{-1} = \begin{bmatrix} \gamma & -v\gamma \\ -v\gamma & \gamma \end{bmatrix}$$

5.7 Energy and Momentum

The momentum is a vector, so in 3 dimensions of space it is a 3-vector and can be resolved into 3 scalar components once a suitable basis has been chosen.

The kinetic energy is a scalar. But both are observer dependent. A particle at rest relative to the observer has momentum that is the zero vector and kinetic energy zero. Should another particle collide with the one at rest and cause both particles to travel away in new directions and speeds, the total momentum vector and the total kinetic energy scalar will be the same before and after the collision, but the values of these quantities are different depending on the observer.

To another inertial observer moving relative to this scene, exactly the same conservation laws will be found to be obeyed, just with different numbers involved. The two observers will disagree over the momenta and energies of the specific particles, and also over the total energy and momentum, as summed separately over both particles.

As before we will take as our unit of distance the light-second, so the speed of light c is 1. In this unit system, energy and mass are fully equivalent, because the famous:

$$E = mc^2$$

becomes:

$$E = m$$

The total energy of a particle is its mass times the Lorentz factor γ:

$$\gamma = \frac{1}{\sqrt{1 - v^2}}$$

In our everyday experience, v is practically zero, as we're expressing it as a fraction of the speed of light, so almost all the energy of a particle is in its mass. The contribution from the kinetic energy is almost non-existent.

But again, this is a frame-dependent quantity, because a particle only has a defined velocity relative to some chosen inertial frame.

If we combine the components of the momentum 3-vector with the energy scalar (the total energy $m\gamma$ discussed above), we get a 4-vector called the 4-momentum. As always in the Minkowski metric, the magnitudes of these objects are related by:

$$m^2 = E^2 - |\mathbf{p}|^2$$

So the 4-momentum has magnitude m, the energy is "temporal" and the 3-momentum is "spatial". Enjoy the symmetry with:

$$s^2 = t^2 - d^2$$

It's interesting that the momentum provides the three spatial components, while energy provides the remaining temporal component (and indeed this is the case: momentum conservation is due to translational symmetry in space, and energy conservation is due to translational symmetry in time.)

As we know that momentum and energy are separately conserved from the viewpoint of any inertial observer, we therefore know that the combined 4-momentum must also be conserved.

The four coordinates are resolved relative to a coordinate system. We had to choose an orientation for the three axes that make up our spatial basis, which is a slice of spacetime. One way to visualise it is to discard one of the space dimensions, so that space is a planar slice through in a 3D spacetime. A given inertial observer regards their spatial slice through spacetime as containing all the events happening "now", making it a "slice of simultaneity".

Each inertial observer will use different coordinates for the 4-momentum, not just because they have a free choice of spatial basis, but also because they each have an event path through spacetime that is momentarily in a specific direction. Each observer, assuming themselves to be at rest (at least instantaneously), regards their own event path as aligned with the time axis, and orthogonal to their slice of simultaneity.

But despite the coordinates of the 4-momentum being different, they describe the same vector in spacetime. That is, two observers stating the 4-momentum of the same particle will use different coordinates for the same 4-momentum. The vector itself can now at last be said to be conserved even under a change of coordinate systems. Everyone agrees on what the 4-momentum is geometrically, so we no longer have to qualify the law of conservation of momentum with caveats about a single frame of reference.

If we scale the 4-momentum by $1/m$ (that is, in some coordinate system, if we divide all the components of the 4-momentum by the particle's intrinsic mass), we obtain the 4-velocity:

$$\mathbf{u} = \frac{\mathbf{p}}{m}$$

The magnitude of this vector is always c (or 1 in our simplified units), because the 4-momentum's magnitude is always mc (or m). In other words, the only information really carried by the 4-velocity is a direction. We're only modelling a *direction* in spacetime and can ignore the magnitude as not physically significant.

This division by m is not meaningful for a massless particle such as a photon, which is why momentum is more fundamental than velocity, as momentum can be discussed for all particles regardless of whether they have mass. Yet it's interesting that a photon's 4-vector nevertheless has magnitude c (or 1), just as if it had a very small mass m that we could divide by.

The relationship between the components of the 4-velocity $\mathbf{u}$ and familiar concepts is not as straightforward as for the 4-momentum. The spatial slice of $\mathbf{u}$ points in the direction of motion (of course), as does the ordinary velocity $\mathbf{v}$. But their magnitudes, u and v, are different:

$$u = \gamma v$$

Where γ is the Lorentz factor again. As we noted above, there's an identical relationship between the total energy E and the intrinsic mass:

$$E = \gamma m$$

A curved surface can be pictured as being embedded in a higher dimensional space, e.g. the surface of the Earth in three dimensional space, and we might ask how else could it be curved if not thanks to being embedded in a space that gives it another degree of freedom besides those spanned by its surface? Nevertheless a theory of curved spacetime has no immediate need of higher dimensions to explain anything and so we make no assumptions. To describe the curvature of a surface around some point we can compare it to a flat surface tangent to the curved surface at that point. This is a linear approximation of the curved surface. We can retain this idea even if we do not allow ourselves to picture an actual flat surface as existing tangent to the curved surface (as that would require a higher dimensional space to exist). Instead we suppose that every point on the surface has an associated scalar value, that is, there is a scalar field. It doesn't matter what that field is, but assume it isn't constant but instead varies smoothly between neighbouring points. It therefore has a gradient at every point, which is a linear approximation of how the field is changing around that point. We can suppose the existence of a directional derivative, which is the dot product of some vector with the gradient. These directional derivatives only tell us (to a linear approximation) how the scalar field would change if we moved incrementally away from the point. But this is enough: the set of directional derivatives associated with a point is a vector space tied to that point, which we equate with the tangent space. Thus we've defined the tangent space associated with every point without needing a higher dimensional space for an actual flat tangent space to extend into.

Appendix A

Greek Alphabet

This is the Greek alphabet. Why not memorise it?

Name	Lower	Upper
Alpha	α	A
Beta	β	B
Gamma	γ	Γ
Delta	δ	Δ
Epsilon	ϵ	E
Zeta	ζ	Z
Eta	η	H
Theta	θ	Θ
Iota	ι	I
Kappa	κ	K
Lambda	λ	Λ
Mu	μ	M
Nu	ν	N
Xi	ξ	Ξ
Omicron	o	O
Pi	π	Π
Rho	ρ	R
Sigma	σ	Σ
Tau	τ	T
Upsilon	υ	Υ
Phi	ϕ	Φ
Chi	χ	Q
Psi	ψ	Ψ
Omega	ω	Ω