Willoughby Seago

**Block 1**

# Engineering Mathematics

24th September 2025

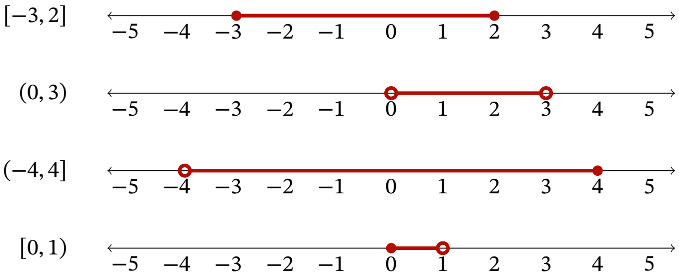
UNIVERSITY OF GLASGOW

# Engineering Mathematics

Willoughby Seago

24th September 2025

These are the lecture notes for block 1 of the *Engineering Mathematics 1* (ENG1063) course. They contain the material covered in the lectures and more. Last updated on October 31, 2025 at 16:31.



# Chapters

# Contents

Page

# List of Figures

Page

# Zero

# Introduction

Welcome to *Engineering Mathematics 1*! These are the lecture notes for block 1 of the course. The notes here should cover all of the content of the lectures, plus some more. The content delivered in lectures is the only *examinable* content. That doesn't mean you should ignore the rest of the material though! Learning the bare minimum amount needed for the exam is *not* a good way to prepare for the exam, and will only hold you back later.

If you find an error in these notes (and I'm sure there will be some) please either contact me via email[1], or create an issue on *Github*[2]. Learning how *Github* works will be very useful if you ever plan to write code (and you will write code at some point).

## 0.1 Notes Format

The notes are *approximately* divided up into one chapter per lecture. The key content is in the definitions and examples. You don't need to remember these word for word, but you should be able to recreate the definitions and reproduce the work that went into doing an example.

> **Definition 0.1.1** Boxes like this will be used to state definitions. You don't need to remember these word for word, but you should be able to give an equivalent definition.

Other definitions are given in the text with the word in **bold** being defined. These are still important definitions to know.

> **Notation 0.1.2** Boxes like this are used to define notation. You are expected to be familiar with this notation.

> **Example 0.1.3** Boxes like this will be used for examples. These may or may not have been covered in the lecture. You don't need to remember the exact details of any example. The examples should be similar to questions that could be asked in an exam, so make sure you *understand* the example.

[1] willoughby dot seago at glasgow dot ac dot uk

[2] https://github.com/WilloughbySeago/engineering-mathematics-lecture-notes

**Application 0.1.4** Boxes like this will be used for applications. These are basically examples but with a bit more context, so the deal is the same: understand them, don't need to memorise them.

**Problem 0.1.5** Boxes like this are used to give problems. You should attempt these, but there's no grade for them. Some may require you to pause and work something out, others you can just think about. There are no answers provided for these, but I'm happy to discuss them.

**Code 0.1.6** Boxes like this will be used for code. This will mostly be *Matlab* code, since you should all learn some *Matlab* during the course. You don't need to memorise or understand this code for the exams, but I find that if I can code something up then I probably understand it well.
I'm not an expert at *Matlab*, so don't trust my code too much!

Boxes like this will contain important ideas!

! Here's a warning, just pointing out something to look out for. This might be an edge case to consider or a common mistake that students make.

**Remark 0.1.7** This is a side comment, it's definitely *not examinable*, and you don't need to understand it. It's just there if you're interested in the maths (and is part of my sneaky plan to convince you all that maths is interesting!). I may also add links to relevant sources (usually just the *Wikipedia* page, most of the time *Wikipedia* is actually very good for maths, if a bit hard to read). You are under no obligation to look at any of these links. I'd be happy to discuss this content with you if you want, but not during lectures, and not if it gets in the way of other students discussing examinable material.

**Theorem 0.1.8.** A theorem is an important result.

*Proof.* Often a theorem is accompanied by a proof, showing that it is true. The details of a proof will not be examinable, in the sense that you won't be expected to recreate the proof. However, understanding the proof can bring insight to how things work and why the result is true.
We usually end a proof with some sort of mark, here the mark is an empty square. Other common marks are filled in squares, or the letters QED, meaning quod erat demonstrandum, Latin for "that which was to be demonstrated", meaning we've shown the thing we set out to show. □

Table 0.1: The Greek alphabet.

| Letter | Lower case | Upper case | Letter | Lower case | Upper case |
|---|---|---|---|---|---|
| Alpha | $\alpha$ | $A$ | Nu | $\nu$ | $N$ |
| Beta | $\beta$ | $B$ | Xi | $\xi$ | $\Xi$ |
| Gamma | $\gamma$ | $\Gamma$ | Omicron | $o$ | $O$ |
| Delta | $\delta$ | $\Delta$ | Pi | $\pi$ or $\varpi$ | $\Pi$ |
| Epsilon | $\varepsilon$ or $\epsilon$ | $E$ | Rho | $\rho$ or $\varrho$ | $P$ |
| Zeta | $\zeta$ | $Z$ | Sigma | $\sigma$ or $\varsigma$ | $\Sigma$ |
| Eta | $\eta$ | $H$ | Tau | $\tau$ | $T$ |
| Theta | $\theta$ or $\vartheta$ | $\Theta$ | Upsilon | $\upsilon$ | $\Upsilon$ |
| Iota | $\iota$ | $I$ | Phi | $\phi$ or $\varphi$ | $\Phi$ |
| Kappa | $\kappa$ or $\varkappa$ | $K$ | Chi | $\chi$ | $X$ |
| Lambda | $\lambda$ | $\Lambda$ | Psi | $\psi$ | $\Psi$ |
| Mu | $\mu$ | $M$ | Omega | $\omega$ | $\Omega$ |

**Lemma 0.1.9** A lemma is a result which is true, but probably only relevant in that it helps us prove other results, rather than being interesting on its own, but there are still some very interesting lemmas!

**Proposition 0.1.10** A proposition is a result which is less important than a theorem (where exactly the line is between theorem and proposition is up to the author).

## 0.2 Symbols and Alphabets

Maths is full of lots of symbols. Any important ones will be defined in the notes. We also like to use other alphabets in maths. The Greek alphabet (Table 0.1) is particularly common. Some upper case letters, as well as lower case omicron, are the same as the corresponding Latin (normal) letters, so we don't use them in maths. There are also some letters with common "variant" forms, which are the same letter but in different fonts. Occasionally people will use both a letter and its variant to mean different things, but this should be avoided, just pick the one you prefer the look of and use that.

## 0.3 Shorthand

I'm liable to use some shorthand to save writing in the lectures. Here's a list of the most common abbreviations I might use (feel free to ask what I mean in the lecture too).

- b/c – because
- $\therefore$ – therefore
- LHS – left hand side
- RHS – right hand side
- w/ – with
- w/o – without

- +ve – positive
- −ve – negative
- # – number
- num – number
- pt(s) – point(s)
- eqn – equation
- st – such that

# Part I

# Sets and Algebra

# One

# Sets

## 1.1 Sets

**Definition 1.1.1 — Set** A **set** is a collection of things. We call the things in the set **elements** of the set.

**Notation 1.1.2** If $X$ is a set then we write $a \in X$ to mean $a$ is an element of $X$. We may also write $a \notin X$ to mean $a$ is *not* an element of $X$.

There are several ways to define a set. The first is to just list all of the elements. We do this in curly brackets:

$$\{1, 2, 3\}, \qquad \{a, \beta, \clubsuit, \mathcal{D}\}, \qquad \{1, \pi, \{42, 57\}\}. \tag{1.1.3}$$

Notice that the elements can be pretty much anything, numbers, symbols, or even other sets, and we can mix and match these in a set. The order of elements is not important, and we ignore any repeats. So all of the following are the same set:

$$\{1, 2, 3\}, \qquad \{2, 1, 3\}, \qquad \{1, 1, 2, 3\}, \qquad \{1, 3, 2, 1, 3, 2, 2, 2\}. \tag{1.1.4}$$

**Remark 1.1.5** This definition – a collection of things – is somewhat vague. Unfortunately giving a precise definition of a set is actually very hard. The state-of-the-art definition is the axioms of Zermelo–Fraenkel (ZF) set theory, which are pretty complicated (possibly with the addition of the axiom of choice for ZFC). They're mostly concerned with edge cases that we don't have to worry about. The only rule we really need to add is that no set can be an element of itself, otherwise we have problems with Russel's paradox.

Two sets are **equal** if they have *exactly* the same elements. That is, if $X$ and $Y$ are sets then $X = Y$ if every element of $X$ is an element of $Y$ and every element of $Y$ is an element of $X$.

Sets can have any number of elements, including zero. The set with zero elements is called the **empty set**, and denoted $\varnothing$ or $\{\}$[1]. Sets can also have an infinite number of elements! The number of elements of a set is called the **cardinality** of the set.

[1] Sometimes the symbols $\emptyset$ or $\phi$ are used also.

Another way to define a set is from an existing set and a condition. We do this using curly brackets to For example, if we have the set $A = \{1, 2, \dots, 10\}$ then we can form new sets using the notation

$$\{a \in A \mid \text{condition on } a\}. \tag{1.1.6}$$

The resulting set is all elements of $A$ which make the condition true. Some texts will use $:$ in place of $|$.

For example,

$$\{a \in A \mid a \text{ is even}\} = \{2, 4, 6, 8, 10\}, \tag{1.1.7}$$
$$\{x \in A \mid x \neq 7\} = \{1, 2, 3, 4, 5, 6, 8, 9, 10\}, \tag{1.1.8}$$
$$\{\alpha \in A \mid 2\alpha \in A\} = \{1, 2, 3, 4, 5\}. \tag{1.1.9}$$

It is important to include which set $a$ comes from, done here with $a \in A$. If you don't then it's not clear which values of $a$ we should try in the condition. You can also write which set $a$ comes from as part of the condition.

### 1.1.1 Special Sets

The following definitions are some sets that it's useful to have a special notation for. These use an alternative font called black board bold, so called because when writing on the board doubling up some lines is about as close to a bold font as you can get. Here's the uppercase alphabet in black board bold for reference:

$$\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}. \tag{1.1.10}$$

This will look slightly different in different fonts. I suggest having a practice writing the black board bold letters used in the following definitions.

**Definition 1.1.11 — Natural Numbers** The **natural numbers** is the set

$$\mathbb{N} = \{1, 2, 3, \dots\} \tag{1.1.12}$$

of all positive whole numbers.

**Remark 1.1.13** Some people (including me) would prefer to define the natural numbers as

$$\mathbb{N} \overset{!}{=} \{0, 1, 2, 3, \dots\}. \tag{1.1.14}$$

However, both the textbook and the chosen convention of the Glasgow university maths courses is that $0 \notin \mathbb{N}$, so that's what we'll go with. This is simply a choice of convention, there's nothing incorrect about either definition, it's just which one is more useful for the maths you're currently doing.

Because of the ambiguity of what $\mathbb{N}$ may mean with these differing conventions it's common to see other notations, such as

$$\mathbb{N}^* = \mathbb{N}^\times = \mathbb{N}_{>0} = \mathbb{Z}_{>0} = \{1, 2, 3, \dots\}; \tag{1.1.15}$$
$$\mathbb{N} \cup \{0\} = \mathbb{N}_0 = \mathbb{Z}_{\geq 0} = \{0, 1, 2, 3, \dots\}. \tag{1.1.16}$$

Don't worry about any symbols you haven't seen before here, but I may occasionally use $\mathbb{Z}_{\geq 0}$ or $\mathbb{Z}_{>0}$.

**Definition 1.1.17 — Integers** The **integers** is the set

$$\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\} \tag{1.1.18}$$

of all whole numbers.

**Remark 1.1.19** The integers are denoted by $\mathbb{Z}$, which comes from the German *zahlen*, which means numbers.

**Definition 1.1.20 — Rational Numbers** The **rational numbers** is the set

$$\mathbb{Q} = \left\{\frac{a}{b} \mid a, b \in \mathbb{Z}, \text{and } b \neq 0\right\} \tag{1.1.21}$$

of all fractions.

**Remark 1.1.22** The rationals are denoted by $\mathbb{Q}$, because they are all quotients, which is just another word for fraction.

Note that $1/2$, $2/4$, $3/6$, and so on all appear as $a/b$ for some choice of $a$ and $b$, but these are all equal, so between them only define one element of $\mathbb{Q}$. An equivalent definition that gets around this overspecification is

$$\mathbb{Q} = \left\{\frac{a}{b} \mid a, b \in \mathbb{Z}, \text{and } \gcd(a, b) = 1\right\}. \tag{1.1.23}$$

Then we get $1/2$, but not $2/4$ or $3/6$ since $\gcd(2, 4) = 2$ and $\gcd(3, 6) = 3$. Here gcd is the **greatest common divisor**, the largest natural number which divides all of the inputs.

**Definition 1.1.24 — Real Numbers** The **real numbers** is the set, $\mathbb{R}$, the elements of which are all points on the number line.

For example, the real numbers contains all of the integers and all of the rationals, but also things like $\pi$, e, and $\sqrt{2}$. Another way of thinking about this is that $\mathbb{Q}$ consists of all numbers which have a repeating decimal expansion (including, for example, 0.5, which is just 0.500000 with 0 repeating forever). Then $\mathbb{R}$ is all numbers including those without a repeating decimal expansion, such as $\pi = 3.1415926\dots$.

**Remark 1.1.25** I've said "all numbers" here, but that's a bit of a circular definition, since when I say number I really mean real number. You'll see in block 2 that there are other "numbers" that aren't real numbers[a]. These

are the complex numbers, denoted $\mathbb{C}$. In fact, there are many sets we can define in maths that we may wish to call "numbers", so be careful when you use the term "number" to specify what you mean by that.
There are several (equivalent) formal definitions of the real numbers which don't have this problem of circular definitions. However, they're pretty hard to understand and even harder to use, so they aren't that helpful for us.

[a] Which isn't to say they aren't "real" in the day-to-day sense of existing (and being useful).

**Definition 1.1.26 — Intervals** An **interval** is a segment of the number line. An interval can either be **open**, **closed**, or **half-open**, depending on whether we include the endpoints or not. Let $a, b \in \mathbb{R}$ with $a \leq b$.

- Open interval between $a$ and $b$:

$$(a, b) = \{x \in \mathbb{R} \mid a < x < b\}. \tag{1.1.27}$$

- Closed interval between $a$ and $b$:

$$[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}. \tag{1.1.28}$$

- Half-open intervals between $a$ and $b$:

$$(a, b] = \{x \in \mathbb{R} \mid a < x \leq b\}, \tag{1.1.29}$$
$$[a, b) = \{x \in \mathbb{R} \mid a \leq x < b\}. \tag{1.1.30}$$

Note that brackets mean we exclude the endpoint and square brackets mean we include it.

We can draw intervals as lines on the number line. When we do the convention is that an empty circle means we leave out the endpoint, and a filled in circle means we include it. See Figure 1.1.

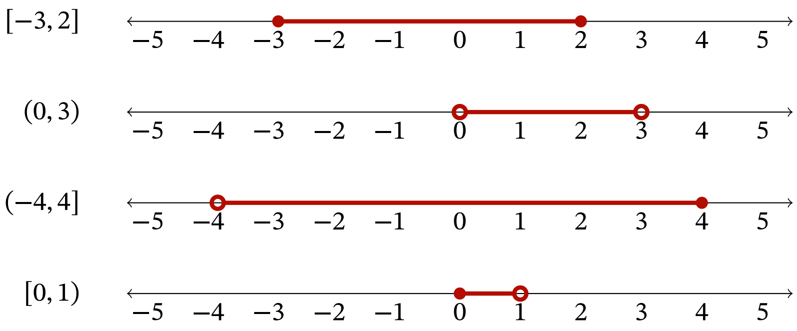


Figure 1.1: Intervals plotted on the number line.

We can also include the symbols $\infty$ and $-\infty$ in our intervals. The rule is that if $x \in \mathbb{R}$ then $x < \infty$, $x \leq \infty$, $x > -\infty$ and $x \geq -\infty$ are always true. However, $\infty$ is *not* a real number, and so it doesn't make sense to include it as an endpoint. We cannot write $[0, \infty]$, but we can write $[0, \infty)$, which is the set

$$[0, \infty) = \{x \in \mathbb{R} \mid 0 \leq x < \infty\}, \tag{1.1.31}$$

which is just the non-negative real numbers.

**Remark 1.1.32** Note that for any $a \in \mathbb{R}$ we have $(a, a) = \{x \in \mathbb{R} \mid a < x < a\} = \varnothing$, there is no number that is both strictly greater than $a$ and strictly less than $a$. So $\varnothing$ is an open interval.

We also have $(-\infty, \infty) = \{x \in \mathbb{R} \mid -\infty < x < \infty\} = \mathbb{R}$, so $\mathbb{R}$ is an open interval.

The fact that $\mathbb{R}$ and $\varnothing$ are both open intervals is important in an area of maths called topology, which generalises the notion of open and closed intervals.

Notice that for any $a \in \mathbb{R}$ we have $[a, a] = \{x \in \mathbb{R} \mid a \leq x \leq a\} = \{a\}$, so any singleton set is a closed interval.

## 1.2 Operations and Orders

### 1.2.1 Operations

**Definition 1.2.1 — Binary Operation** Let $S$ be a set. A binary operation, say $*$, on $S$ takes in two elements, $a, b \in S$, and outputs another element, $a * b \in S$.

Note that we're just using $*$ as symbol here for a general binary operation. Other symbols, such as $+$, $-$, $\times$, $\cdot$, $\circ$, or even no symbol (e.g., just writing $ab$ for the product) are often used.

**Example 1.2.2** The following define binary operations on $\mathbb{R}$:

- $a * b = a + b$;
- $a * b = a - b$;
- $a * b = ab$;
- $a * b = \max\{a, b\}$;
- $a * b = (a + b)/2$;
- $a * b = 14$.

Whenever we have a binary operation there are two properties that we usually want to check for. Not every binary operation has these properties, but when they do they are often particularly nice, so it's always useful to know.

The first is commutativity, which says that the order doesn't matter.

**Definition 1.2.3 — Commutative** A binary operation, $*$, on $S$ is called **commutative** if $a * b = b * a$ for all $a, b \in S$.

**Remark 1.2.4** You may also hear the term “abelian” used to describe a commutative operation. This is named for the mathematician Niels Henrik Abel. This phrase is typically used when $S$ equipped with the binary operation forms a group (don’t worry if you don’t know what a group is).

**Example 1.2.5** Addition on $\mathbb{R}$ is commutative: $x+y = y+x$ for all $x, y \in \mathbb{R}$.
Subtraction on $\mathbb{R}$ is noncommutative: $5-2 = 3$ and $2-5 = -3$. Note that it’s enough to provide a counterexample (here 5 and 2) to show that an operation isn’t commutative, but to show it is commutative you have to show that the order doesn’t matter for all possible inputs.
Multiplication on $\mathbb{R}$ is also commutative.
If you’re familiar with matrices note that matrix multiplication is noncommutative. Another example of a noncommutative operation you may be familiar with is the cross product (or vector product) of two vectors.

**Problem 1.2.6** Are the other operations of Example 1.2.2 commutative?

The other condition is associativity, which says that if we do the operation multiple times it doesn’t matter how we put brackets around it.

**Definition 1.2.7 — Associative** A binary operation, $*$, on $S$ is called **associative** if $(a * b) * c = a * (b * c)$ for all $a, b, c \in S$.

When an operation is associative we usually don’t bother putting the brackets in since it doesn’t matter where we put them. Note that the definition of associativity only uses three elements, but it actually means that for any number of elements where we put the brackets is not important.

**Example 1.2.8** Addition on $\mathbb{R}$ is associative: $(x + y) + z = x + (y + z)$.
Subtraction on $\mathbb{R}$ is not associative: $(5-2)-3 = 3-3 = 0$ and $5-(2-3) = 5-(-1) = 6$.
Multiplication on $\mathbb{R}$ is also associative.
If you’re familiar with matrices note that matrix multiplication is associative. The vector cross product is nonassociative.

**Problem 1.2.9** Are the other operations of Example 1.2.2 commutative?

### 1.2.2 Orders

An order is similar to a binary operation, in that it takes in two elements of some set, $S$. However, the output isn't another value of $S$, but instead the statement is either true or false. For example, $1 < 3$ is true, and $3 < 1$ is false.

There is also a natural way to order sets, and that's by subset.

**Definition 1.2.10 — Subset** A set, $X$, is a **subset** of a set, $Y$, if every element of $X$ is also an element of $Y$. In symbols, if $a \in X$ then $a \in Y$.
We say that $Y$ is a **superset** of $X$ if $X$ is a subset of $Y$.
If $X \neq Y$ and $X$ is a subset of $Y$ then we say $X$ is a **proper subset** of $Y$, and $Y$ is a **proper superset** of $X$. The word **strict** may also be used instead of proper.

Note that this is similar to the definition of when two sets are equal, but without the "exactly". There can be elements of $Y$ which are not elements of $X$. In fact, a common way to show that two sets, $X$ and $Y$, are equal is to show that $X \subseteq Y$ and $Y \subseteq X$.

Nowhere in the definition does it say that $X$ needs to have elements. If $X = \varnothing$ then it is true that every element of $X$ is an element of $Y$, it's just that there are no elements of $X$. Thus, the empty set is a subset of all sets, $\varnothing \subseteq Y$.

**Remark 1.2.11** The empty set satisfies any property which can be stated as "such and such is true for all elements of $X$". We say that the property holds vacuously. For example, if I have an empty field it is true to say that every horse in the field is purple!

**Notation 1.2.12** If $X$ is a subset of $Y$ we write $X \subseteq Y$ or $Y \supseteq X$. If $X$ is a proper subset of $Y$ we write $X \subset Y$ or $Y \supset X$.

! Some sources write $\subset$ to mean subset and $\subsetneq$ to mean proper subset, so be careful.

**Example 1.2.13** Can you see why each of the following is true? Note that $/$ is used to mean that the statement without the $/$ is false.

- $\{1, 2, 3\} \subset \{1, 2, 3, 4\}$;
- $\{1, 2, 3\} \subseteq \{1, 2, 3, 4\}$;
- $\{1, 2, 3\} \subseteq \{1, 2, 3\}$;
- $\{1, 2, 3\} \not\subset \{1, 2, 3\}$;
- $\{1, 2, 3, 4\} \not\subseteq \{1, 2, 3\}$;
- $\{1, 2, 3, 4\} \not\subset \{1, 2, 3\}$.

**Example 1.2.14** Notice that we have a chain of inclusions:

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}. \tag{1.2.15}$$

Can you come up with an element of each set which was not in the previous set, showing that these are strict subsets? If you know about the complex numbers already then note that we can extend this by $\mathbb{R} \subset \mathbb{C}$.

**Problem 1.2.16** Can you list all subsets of $\{1\}$, $\{1, 2\}$, $\{1, 2, 3\}$, and $\{1, 2, 3, 4\}$?
Hint: don't forget the empty set and the whole set.
Can you spot a pattern in the number of subsets?

We can think of $\subseteq$ as defining an order on sets, just like $\leq$ is an order on $\mathbb{R}$. One difference is that for any two real numbers, $x$ and $y$, we always have either $x \leq y$ or $y \leq x$ (or both if $x = y$). However, for sets this isn't the case. For example, if $X = \{1, 2, 3\}$ and $Y = \{3, 4, 5\}$ then it isn't true that $X \subseteq Y$, since $1 \notin Y$, and it isn't true that $Y \subseteq X$, since $4 \notin X$.

**Remark 1.2.17** The difference highlighted above is the difference between a total order and a partial order. The real numbers with $\leq$ are a total order (in fact, this can be taken as one of the defining properties of $\mathbb{R}$), whereas sets are only partially ordered by $\subseteq$.

### 1.2.3 Operations on Sets

In this section let $A$ and $B$ be sets.

**Definition 1.2.18 — Union** The **union** of $A$ and $B$ is the set, $A \cup B$, containing all elements of either $A$ *or* $B$. In symbols,

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}. \tag{1.2.19}$$

**Example 1.2.20**

- $\{1, 2, 3\} \cup \{4, 5, 6\} = \{1, 2, 3, 4, 5, 6\}$;
- $\{1, 2, 3\} \cup \{2, 3, 4\} = \{1, 2, 3, 4\}$;
- $\{1, 2, 3\} \cup \varnothing = \{1, 2, 3\}$;
- $\mathbb{N} \cup \mathbb{Z} = \mathbb{Z}$;
- $\mathbb{N} \cup \{0\} = \{0, 1, 2, 3, \dots\} = \mathbb{Z}_{\geq 0}$.

Notice that the union of two sets needn't be a new set. In particular, if $A$ is a subset of $B$ then $A \cup B = B$.

**Definition 1.2.21 — Intersection** The **intersection** of $A$ and $B$ is the set, $A \cap B$, containing al elements of *both A and B*. In symbols,

$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}. \tag{1.2.22}$$

**Example 1.2.23**

- $\{1, 2, 3\} \cap \{4, 5, 6\} = \varnothing$;
- $\{1, 2, 3\} \cap \{2, 3, 4\} = \{2, 3\}$;
- $\mathbb{R} \cap \mathbb{Q} = \mathbb{Q}$;
- $\mathbb{Z} \cap \{x \in \mathbb{R} \mid -3 \leq x \leq 3\} = \{-3, -2, -1, 0, 1, 2, 3\}$.

Notice that the intersection of two sets needn't be a new set. In particular, if $A$ is a subset of $B$ then $A \cap B = A$.

**Definition 1.2.24 — Difference** The **difference** of $A$ and $B$ is the set, denoted $A \setminus B$ or $A - B$, containing all elements of $A$ which are *not* elements of $B$. In symbols,

$$A \setminus B = \{x \in A \mid x \notin B\}. \tag{1.2.25}$$

**Example 1.2.26**

- $\{1, 2, 3, 4, 5\} \setminus \{4, 5\} = \{1, 2, 3\}$;
- $\mathbb{R} \setminus \mathbb{Q}$ is the **irrational numbers**, all numbers which don't have a repeating decimal expansion;
- $\mathbb{Z} \setminus \mathbb{N} = \{\dots, -3, -2, -1, 0\}$;
- $\mathbb{Z}_{\geq 0} \setminus \{0\} = \mathbb{N}$.

All of these ways of combining sets can be pictured using Venn diagrams (Figure 1.2).

When the sets in question are intervals we can also draw them on the number line to compute the union, intersection, and difference (Figure 1.3). The union is anywhere there's a line. The intersection is anywhere the lines overlap. The difference leaves a hole in the first interval where the second interval is.

The intersection of two intervals is always an interval, but the union and difference of two intervals isn't necessarily an interval, sometimes there's a hole. We can still write the result as a union of intervals though.

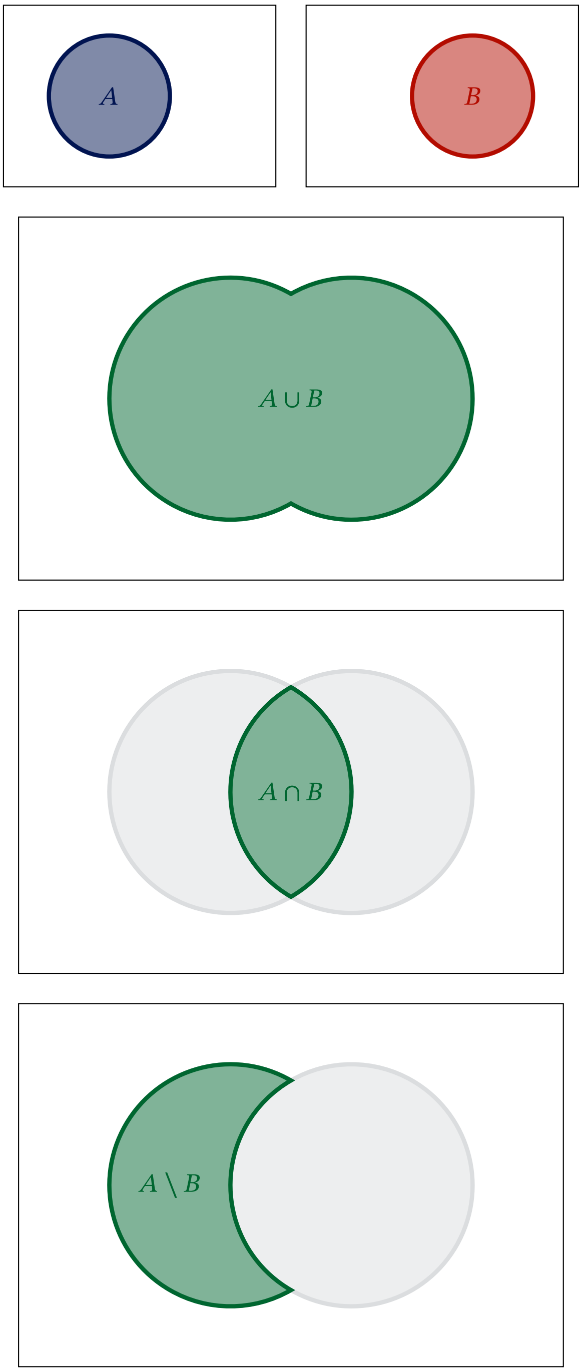


Figure 1.2: The union, intersection, and set difference of the sets $A$ and $B$ represented as Venn diagrams.

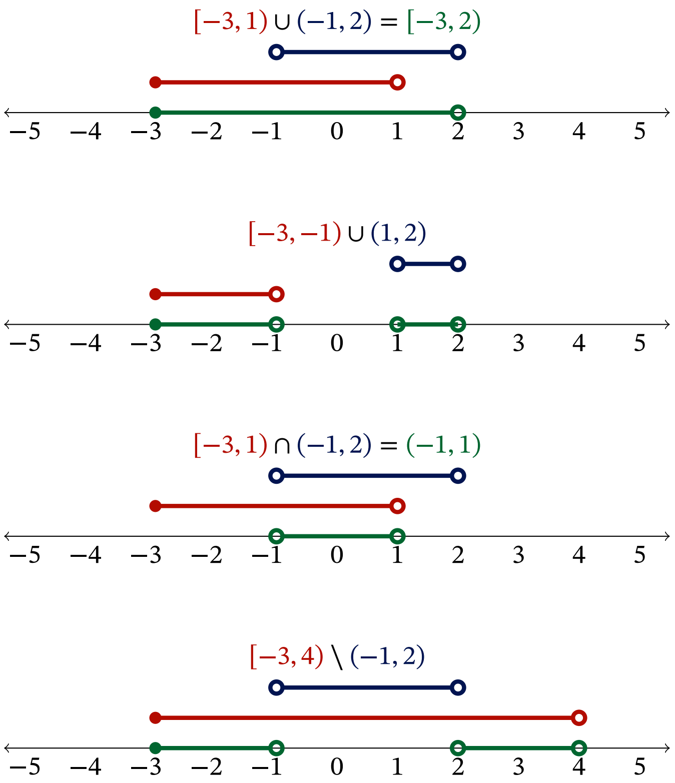


Figure 1.3: Union, intersection, and set difference of intervals. Note that even when the result is made of two different line segments it's still all one set.

## 1.3 Power Rules

Let $a \in \mathbb{R}$ be positive. For $n \in \mathbb{N}$ we define[2]

$$a^n := \underbrace{a \cdot a \cdots a}_{n \text{ factors}}. \tag{1.3.1}$$

From this definition we can derive the first power rule, specifically,

$$a^n a^m = a^{n+m}. \tag{1.3.2}$$

To see this we simply write out the definitions:

$$a^n a^m = \underbrace{a \cdots a}_{n \text{ factors}} \cdot \underbrace{a \cdots a}_{m \text{ factors}} = \underbrace{a \cdots a}_{n+m \text{ factors}} = a^{n+m}. \tag{1.3.3}$$

Often in maths we have a definition that we want to extend in some way. In this case, what if we want to define $a^0$? A good way to do this is to look at what results hold for that definition, and make the extended definition in such a way that these properties still hold[3]. In this case we have that $a^n a^m = a^{n+m}$. If we take $m = 0$ then we should have $a^n a^0 = a^{n+0} = a^n$. We can see that if we define

$$a^0 := 1 \tag{1.3.4}$$

then this result is still true, so that's the definition we'll take.

[2] The symbol := is sometimes used to mean that the left-hand-side is *defined* to be the same as the right-hand-side.

[3] The other way results get generalised in maths is pretty much the opposite of this, we ask instead what would happen if we deliberately break a property that holds in the more restricted case.

We can continue on with this. If we want to define $a^{-n}$ for $n \in \mathbb{N}$ then we should define it in such a way that the equation $a^n a^{-n} = a^{n+(-n)} = a^0 = 1$ holds. That is, we should make the definition

$$a^{-n} := \frac{1}{a^n}. \tag{1.3.5}$$

Another property that we can check holds for $n, m \in \mathbb{N}$ is

$$(a^n)^m = a^{nm}. \tag{1.3.6}$$

To see this holds we again just write out the definitions:

$$(a^n)^m = \underbrace{a^n \cdots a^n}_{m \text{ factors}} = \underbrace{\underbrace{a \cdots a}_{n \text{ factors}} \cdot \underbrace{a \cdots a}_{n \text{ factors}}}_{m \text{ factors}} = \underbrace{a \cdots a}_{nm \text{ factors}} = a^{nm}. \tag{1.3.7}$$

Next we ask how we should define $a^{1/n}$. If we still want this property to hold we should have $(a^{1/n})^n = a^{n/n} = a^1 = a$. That is, we should define $a^{1/n}$ to be the number whose $n$th power is $a$. If that's a bit confusing just consider $n = 2$. Then $a^{1/2}$ should be the number which squares to $a$. That is, $a^{1/2} = \sqrt{a}$. More generally, we make the definition

$$a^{1/n} := \sqrt[n]{a}. \tag{1.3.8}$$

**Remark 1.3.9** There's a slight subtlety here about exactly what we mean by $\sqrt{a}$ or $\sqrt[n]{a}$. For example, both 2 and $-2$ square to give 4. When $a$ is a positive real number we will always mean that $\sqrt[n]{a}$ is the *positive* real number whose $n$th power is $a$. When $a$ is negative or even complex then we have to be more careful.

For ease of use here are all of the results of this section in one place. For $a$ a positive real number and $m, n \in \mathbb{N}$ we have

$$a^n a^m = a^{n+m}, \quad a^0 = 1, \quad a^{-n} = \frac{1}{a^n}, \quad \text{and} \quad a^{1/n} = \sqrt[n]{a}. \tag{1.3.10}$$

Note that these can all be combined, for example,

$$a^{n/m} = \sqrt[m]{a^n} = (\sqrt[m]{a})^n. \tag{1.3.11}$$

# Two

# Equations and Inequalities

## 2.1 Absolute Value

Sometimes we want to "throw away" the sign of a quantity. To do so we make the following definition. We use a piecewise definition, which lists the output and then the condition when that output applies:

$$\begin{cases} \text{output 1} & \text{condition 1;} \\ \text{output 2} & \text{condition 2;} \\ \vdots & \vdots\,. \end{cases} \tag{2.1.1}$$

Make sure to cover all cases when you do this.

**Definition 2.1.2 — Absolute Value** For $x \in \mathbb{R}$ we define the **absolute value** of $x$ to be the quantity

$$|x| := \begin{cases} x & \text{if } x \geq 0; \\ -x & \text{if } x < 0. \end{cases} \tag{2.1.3}$$

**Example 2.1.4** What is $|3|$? Well, $3 \geq 0$, so $|3| = 3$.
What is $|-5|$? Well, $-5 < 0$, so $|-5| = -(-5) = 5$.

This is plotted in Figure 2.1.

The idea here is that $|x|$ is the distance from 0 to $x$, it doesn't matter which side of the number line $x$ is on, the distance is $|x|$. For example, both 2 and $-2$ are a distance[1] 2 from 0.

The absolute value is multiplicative, that is, if $x, y \in \mathbb{R}$ then

$$|x||y| = |xy|. \tag{2.1.5}$$

Think about it, the sign of $x$ and $y$ in $xy$ only controls which side of zero $xy$ is on, not how far away it is. For example, $2 \cdot 5 = (-2)(-5) = 10$ and $2(-5) = (-2)5 = -10$, however we add minus signs the result is always 10 away from the origin.

Another property is slightly less obvious, it's called the **triangle inequality**, it states that for $x, y \in \mathbb{R}$ we have

$$|x + y| \leq |x| + |y|. \tag{2.1.6}$$

[1] On the number line there are no units, but in real life we probably want distances to have units.

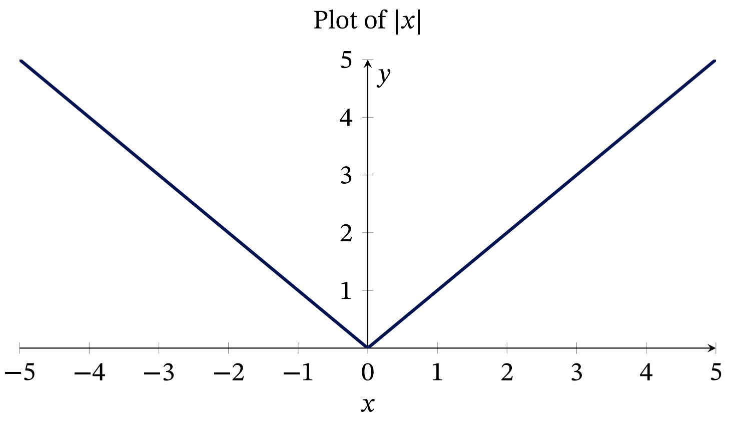


Figure 2.1: Plot of $y = |x|$.

To see this notice that if we want to get as far away from 0 as possible then both $x$ and $y$ should have the same sign. In this case we get equality above. If the signs are different then $x + y$ will always be closer to 0.

**Remark 2.1.7** This result is called the triangle inequality because the same result is true when we measure distances in the plane. There $\mathbf{x}$ and $\mathbf{y}$ are vectors and $|\mathbf{x}|$ and $|\mathbf{y}|$ are the distance of these points from $\mathbf{0} = (0, 0)$. The triangle comes from the definition of adding vectors, joining them tip-to-tail (Figure 2.2), and completing the triangle. The resulting vector's length is always at most as long as the lengths of the other two vectors combined, and it only achieves this length when both $\mathbf{x}$ and $\mathbf{y}$ point in the same direction.

Our case is just the one-dimensional version of this, where direction is just indicated by a sign.

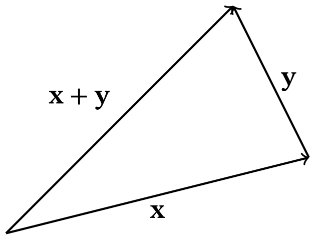


Figure 2.2: The triangle inequality: $|\mathbf{x} + \mathbf{y}| \leq |\mathbf{x}| + |\mathbf{y}|$.

**Remark 2.1.8** The notion of a distance satisfying the triangle inequality generalises to the notion of a metric space. There are some other requirements too: distance should always be positive, the distance of something from itself should be zero, the distance between two different things is positive, and it doesn't matter if we measure from $x$ to $y$ or $y$ to $x$, the distance should be the same.

## 2.2 Inequalities

We can solve inequalities, just like we can solve equations, by finding the *set* of all possible solutions. The only thing to be careful about is that if we multiply or divide both sides of an inequality by a negative number then we need to "flip the inequality". So, $\leq$ becomes $\geq$ and $<$ becomes $>$. To see why this is true just notice that $3 < 5$ and $-3 > -5$.

The following example shows how we can use sets, particularly intervals, to find the solution sets of algebraic inequalities. Note that often it's easier to leave things in terms of inequalities until the end, and only then turn the answer into a set.

**Example 2.2.1** Find the set of all $x \in \mathbb{R}$ satisfying

$$\frac{1}{3-x} < 2. \tag{2.2.2}$$

We can split into three solutions, depending on whether $3 - x$ is positive, negative, or zero.

1. If $3-x = 0$ then we're dividing by 0, which isn't allowed, so we must exclude $x = 3$ from our final solution set.
2. If $3 - x > 0$ then we must have that $3 > x$. We can then multiply by $3 - x$ giving

   $$1 < 2(3 - x) = 6 - 2x. \tag{2.2.3}$$

   Then we can add subtract 6 from both sides giving

   $$-5 < -2x. \tag{2.2.4}$$

   Dividing by $-2$, and remembering to flip the inequality, we have

   $$\frac{5}{2} > x. \tag{2.2.5}$$

   So, the solution for this case is that $x < 3$ and $x < 5/2$ (note $5/2 = 2.5 < 3$). Both can be true at once, and in particular for both to be true we need to have $x < 5/2$. We can turn $x < 3$ and $x < 5/2$ into the interval notation $x \in (-\infty, 3)$ and $x \in (-\infty, 5/2)$. The solution for this case is then the intersection $(-\infty, 3) \cap (-\infty, 5/2) = (-\infty, 5/2)$.
3. If $3 - x < 0$ then we must have that $3 < x$. We can then multiply by $3 - x$ and flip the inequality, giving

   $$1 > 6 - 2x. \tag{2.2.6}$$

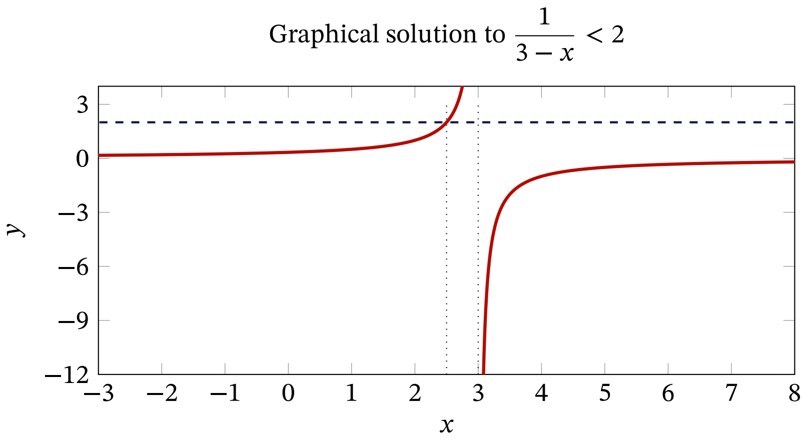


Figure 2.3: Graphical solution to $1/(3-x) < 2$. The horizontal line is $y = 2$, and the curve is $y = 1/(3-x)$. Only between the vertical dashed lines at $x = 5/2$ and $x = 3$ is the curve above $y = 2$. Note that there's a horizontal asymptote at $y = 0$, so the curve never rises up to cross $y = 2$ again on the right.

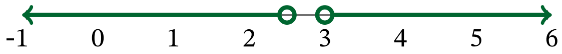


Figure 2.4: Solution set of $1/(3-x) < 2$, which is $(-\infty, 5/2) \cup (3, \infty)$.

Subtracting 6, dividing by $-2$, and flipping the inequality again we get

$$\frac{5}{2} < x. \tag{2.2.7}$$

So we have $x > 3$ and $x > 5/2$, or $x \in (3, \infty)$ and $x \in (5/2, \infty)$. Both conditions must be true, so the solution set is the intersection: $(3, \infty) \cap (5/2, \infty) = (3, \infty)$.

So if $x$ is in either $(-\infty, 5/2)$ or $(3, \infty)$ as long as $x \neq 3$ we have a solution. Thus, the solution set is $\big((-\infty, 5/2) \cup (3, \infty)\big) \setminus \{3\} = (-\infty, 5/2) \cup (3, \infty)$. Note that 3 wasn't actually in either solution set here, so removing it doesn't change anything. This won't always be the case. It may be more familiar to state the solution as $x < 5/2$ or $x > 3$, but really we should state what sort of object $x$ is, a real number, so the solution set is $\{x \in \mathbb{R} \mid x < 5/2 \text{ or } x > 3\}$, which is exactly $(-\infty, 5/2) \cup (3, \infty)$.
Figure 2.3 shows how we can plot $y = 1/(3-x)$ and $y = 2$ to solve this graphically. There we see that between 5/2 and 3 the graph is at or above $y = 2$, so our solution should be $\mathbb{R} \setminus [5/2, 3] = (-\infty, 5/2) \cup (3, \infty)$.
The solution set is plotted on the number line in Figure 2.4.

**Example 2.2.8** Find the set of all $x \in \mathbb{R}$ satisfying

$$\frac{x-2}{x+1} > 4. \tag{2.2.9}$$

If we were solving an equality we would start by multiplying by $x + 1$, but we have to be careful, because $x + 1$ may be negative. We'll split into cases:

- If $x + 1 = 0$ then we're dividing by 0, which isn't allowed. So we manually exclude $x = -1$ from the final result.
- If $x + 1$ is positive then $x + 1 > 0$, so $x > -1$. Then we want to solve

  $$x - 2 > 4(x + 1) = 4x + 4. \tag{2.2.10}$$

  Subtracting $x$ from both and subtracting 4 from both sides we get

  $$-6 > 3x. \tag{2.2.11}$$

  Dividing by 3 we get

  $$-2 > x. \tag{2.2.12}$$

  We see that in this case we need $x > -1$ and $x < -2$, which can't both be true, so this case doesn't contribute any solutions (but we still needed to check it!). The solution set from this case is $\varnothing$.
- If $x + 1$ is negative then $x + 1 < 0$, so $x < -1$. We can multiply by $x + 1$, flipping the inequality as we do, giving

  $$x - 2 < 4(x + 1) = 4x + 4. \tag{2.2.13}$$

  Subtracting $x$ and 4 from both sides we get

  $$-6 < 3x. \tag{2.2.14}$$

  Dividing by 3 we get

  $$-2 < x. \tag{2.2.15}$$

  So we need to have $x > -2$ and $x < -1$ at the same time. This means the solution set is the interval $(-2, -1)$.

The full solution set is then the union of the solution sets of each case. So it's $\varnothing \cup (-2, -1) = (-2, -1)$, and note that $-1$ is not in the solution so we don't need to remove it. It may be more familiar to state the solution as $-2 < x < -1$, but we should really specify what sort of thing $x$ is, a real number, so we should give the solution set as $\{x \in \mathbb{R} \mid -2 < x < -1\}$, which is exactly the interval $(-2, -1)$.

Figure 2.5 shows how we can plot $y = (x - 2)/(x + 1)$ and $y = 4$ to solve this graphically. There we see that between $-2$ and $-1$ the graph is at or above $y = 4$, so our solution should be $(-2, -1)$. Note that we want the graph to be strictly above $y = 4$, so we don't include the endpoints.

The solution set is plotted on the number line in Figure 2.6.

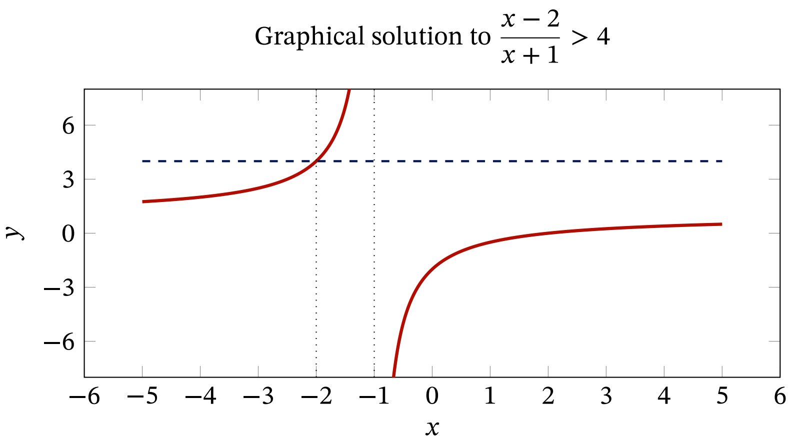


Figure 2.5: Graphical solution to $(x-2)/(x+1) > 4$. The horizontal line is $y = 4$, and the curve is $y = (x-2)/(x+1)$. Only between the vertical dashed lines at $x = -2$ and $x = -1$ is the curve above $y = 4$. Note that there's a horizontal asymptote at $y = 1$, so the curve never rises up to cross $y = 4$ again on the right.

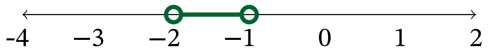


Figure 2.6: Solution set of $(x-2)/(x+1) > 4$, which is $(-2,-1)$.

You'll see from these examples that plotting things can be very useful, at least to check your answers. Making these plots by hand would require that you solve these inequalities. Fortunately, we can often use a computer to make our plots for us. Have a go at plotting these in something like Desmos. Or if you know a little bit of programming you could use Matplotlib and Python, Matlab, or your preferred language with plotting capabilities. Notice that I still used the answer to plot the vertical lines, but you could estimate them from the graph, or use some more advanced code to compute them for you.

**Code 2.2.16** Here's some Matlab code to plot $y = (x-2)/(x+1)$ and $y = 4$. The output is Figure 2.7.

```
x1 = linspace(-5, -1.1, 100);
x2 = linspace(-0.9, 5, 100);

function y = f(x)
    y = (x - 2) ./ (x + 1);
end

hold on
axis([-5, 5, -2, 5])
plot(x1, f(x1), Color="r")
plot(x2, f(x2), Color="r")
plot([-5, 5], [4, 4], "b--")
```

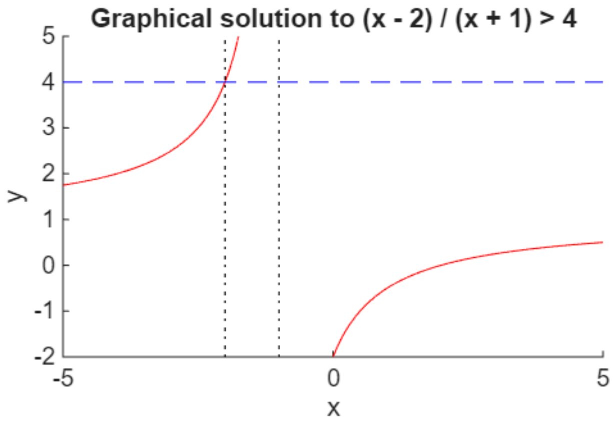


Figure 2.7: The output of Code 2.2.16.

```
plot([-2, -2], [-2, 5], "k:", Marker="none")
plot([-1, -1], [-2, 5], "k:", Marker="none")
title("Graphical solution to (x - 2) / (x + 1) > 4")
xlabel("x")
ylabel("y")
```

**Example 2.2.17** Find the set of all $x \in \mathbb{R}$ satisfying

$$|3x + 6| + x < 4. \tag{2.2.18}$$

We consider cases, $3x + 6 \geq 0$ and $3x + 6 < 0$:

1. If $3x + 6 \geq 0$ then $3x \geq -6$ and $x \geq -2$. In this case we have $|3x + 6| = 3x + 6$, and so we have

   $$3x + 6 + x < 4 \tag{2.2.19}$$

   which we can solve to find

   $$x < -\frac{1}{2}. \tag{2.2.20}$$

   Combining these we have that $-2 \leq x < -1/2$. As a set, $x \in [-2, -1/2)$.

2. If $3x + 6 < 0$ then $3x < -6$ and $x < -1/2$. In this case we have $|3x + 6| = -(3x + 6)$, and so we have

   $$-3x - 6 + x < 4 \tag{2.2.21}$$

   which we can solve (remembering to flip the inequality when we divide by a negative) to find

   $$x > -5. \tag{2.2.22}$$

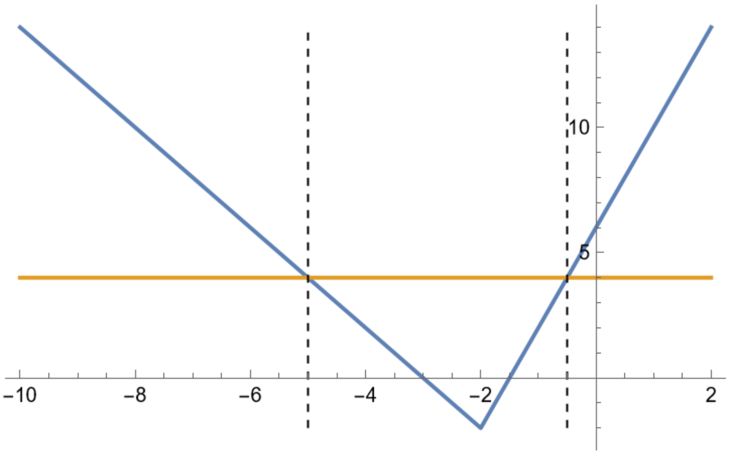


Figure 2.8: The output of Code 2.2.23

> Combining these we have that $-5 < x < -1/2$. As a set, $x \in (-5, -1/2)$.
>
> Since either case is acceptable we want $x \in [-2, -1/2)$ or $x \in (-5, -1/2)$. The solution set is therefore the union $[-2, -1/2) \cup (-5, -1/2) = (-5, -1/2)$.

**Code 2.2.23** Here's some code plotting $y = |3x + 6| + x$ and $y = 4$ in Mathematica. Here I use `Solve` to find the intersection points, then plot the graph with `Plot` and plot the vertical lines with `Line`. The `Show` and `Graphics` commands just make everything appear on the same plot. The output is Figure 2.8.

```
intersectx = x /. Solve[Abs[3 x + 6] + x == 4];
Show[{
    Plot[{Abs[3 x + 6] + x, 4}, {x, -10, 2}],
    Graphics[{Dashed,
        Line[{{intersectx[[1]], -2},
            {intersectx[[1]], 14}}],
        Line[{{intersectx[[2]], -2},
            {intersectx[[2]], 14}}]}]
}]
```

## 2.3 Quadratics

A **quadratic equation** is an equation of the form

$$ax^2 + bx + c = 0. \tag{2.3.1}$$

Here $x$ is a variable and the coefficients, $a$, $b$, and $c$ are some sort of numbers. We'll assume our coefficients are real numbers, but sometimes it makes sense to restrict to integers, and in the next block you'll see that often it's useful to extend to complex numbers.

The goal is to find all values of $x$ which make this equation true. If we restrict $x$ to be a real number then it turns out that such an equation has either 0, 1, or 2 solutions. This follows from the quadratic equation, which is our first method for solving quadratics.

### 2.3.1 Quadratic Formula

The **quadratic formula** provides the solution(s), $x$, to the quadratic equation of Equation (2.3.1). The solution(s) are

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{2.3.2}$$

Notice the square root. If $x$ is to be a real number we can only take square roots of non-negative quantities. We call $\Delta = b^2 - 4ac$ the **discriminant** of the quadratic. It helps us tell the difference between which case we're in, 0, 1 or 2 solutions:

- If $\Delta > 0$ then $x = (-b + \sqrt{\Delta})/2a$ and $x = (-b - \sqrt{\Delta})/2a$ are two distinct real solutions.
- If $\Delta = 0$ then $x = -b/2a$ is the only solution, you'll also hear this being called a repeated root (root just being another word for the solution to an equation). It's as if this solution somehow appears twice, we'll see why in the next section on factorisation.
- If $\Delta < 0$ then we can't take the square root, and so there are no real solutions. You'll see in the next block that there are *complex* solutions still. In fact, if we allow complex roots then there are always two solutions, so long as we count the repeated solutions of the $\Delta = 0$ case as two solutions (which is why we say 2 *distinct* solutions for $\Delta > 0$).

**Example 2.3.3** Solve

$$3x^2 + x - 2 = 0 \tag{2.3.4}$$

using the quadratic equation.

We simply identify $a = 3$, $b = 1$, and $c = -2$. We then have $\Delta = b^2 - 4ac = 1^2 - 4 \cdot 3 \cdot (-2) = 25$, which is positive, so we expect two distinct solutions. Plugging these values into the equation we find the solutions are

$$x = \frac{-1 \pm \sqrt{25}}{2 \cdot 3} \tag{2.3.5}$$

which gives the solutions

$$x = \frac{-1-5}{6} = -1, \qquad \text{and} \qquad x = \frac{-1+5}{6} = \frac{2}{3}. \tag{2.3.6}$$

## 2.3.2 Factorising

When the roots of a quadratic aren't too horrible it is often possible to factorise it. Then the roots are simply the values of $x$ which make each term in the factorisation vanish.

**Example 2.3.7** Solve

$$3x^2 + x - 2 = 0 \tag{2.3.8}$$

by factorising.
The factorisation process is a bit of an art. We'll assume that there are no fractions appearing as coefficients of $x$ in the formula (you can always multiply by any denominator that appears to get rid of it). Then the factorisation must be of the form

$$(3x + \alpha)(x + \beta) = 0 \tag{2.3.9}$$

for some $\alpha, \beta \in \mathbb{R}$. There are several methods for finding $\alpha$ and $\beta$. One is just to stare at this for a while until you can see the solution. Another is to expand these brackets and equate coefficients, so let's do that. Expanding the brackets we get

$$3x^2 + \alpha x + 3\beta x + \alpha\beta = 3x^2 + (\alpha + 3\beta)x + \alpha\beta. \tag{2.3.10}$$

Equating coefficients we have that $\alpha + 3\beta = 1$ and $\alpha\beta = -2$. These are simultaneous equations, which can also be solved in many ways. The second equation tells us that $\beta = -2/\alpha$, which we can substitute into the first, giving

$$\alpha - \frac{2}{3}\alpha = 1 \implies \frac{1}{3}\alpha = 1 \implies \alpha = 3. \tag{2.3.11}$$

Then we have $\beta = -2/3$. This gives

$$(3x + 3)\left(x - \frac{2}{3}\right) = 0. \tag{2.3.12}$$

For this to be true it must be that either

$$3x + 3 = 0, \qquad \text{or} \qquad x - \frac{2}{3} = 0. \tag{2.3.13}$$

Solving these equations we have

$$x = -1, \qquad \text{or} \qquad x = \frac{2}{3}. \tag{2.3.14}$$

Consider the quadratic $x^2 - 2x + 1$. This has $\Delta = (-2)^2 - 4 \cdot 1 \cdot 1 = 0$ and factorises as $(x - 1)^2$. The two factors of $x - 1$ are why we call $x = 1$ a repeated root of this quadratic.

### 2.3.3 Completing The Square

The quadratic

$$ax^2 + bx + c = 0 \tag{2.3.15}$$

can always be written as

$$a\left(x + \frac{b}{2a}\right)^2 - \frac{b^2}{4a} + c = ax^2 + bx + c, \tag{2.3.16}$$

which you can check by expanding the left hand side. The process of doing so is called **completing the square**.

I advise that you *don't* memorise this formula. Instead just practice with specific quadratics and you'll learn the process for completing the square.

While completing the square is usually not the fastest way to solve a quadratic equation it can be useful if you're trying to plot a quadratic, since it's generally easier to plot a quadratic of the form $(x - p)^2 + q = 0$, since the turning point of this quadratic has a turning point at $(p, q)$. Be careful about signs when you do this.

**Example 2.3.17** Solve

$$3x^2 + x - 2 = 0 \tag{2.3.18}$$

by completing the square.
First factorise out the coefficient of $x^2$ from the $x^2$ and $x$ terms, giving

$$3(x^2 + x/3) - 2 = 0. \tag{2.3.19}$$

Our goal is to write $x^2 + x/3$ in the form $(x + p)^2 + q$ for some $p$ and $q$. To do this I like to equate coefficients, expanding we have

$$(x + p)^2 + q = x^2 + 2px + p^2 + q = x^2 + \frac{1}{3}x. \tag{2.3.20}$$

Equating coefficients we have $2p = 1/3$, so $p = 1/6$. We also have $p^2 + q = 0$, so $q = -1/36$. Then we have

$$3\left(\left(x + \frac{1}{6}\right)^2 - \frac{1}{36}\right) - 2 = 0. \tag{2.3.21}$$

Expanding the outer brackets this becomes

$$3\left(x + \frac{1}{6}\right)^2 - \frac{25}{12} = 0. \tag{2.3.22}$$

At this point it's a good idea to expand fully and check that you get $3x^2 + x - 2$ back.
Now that we have this form we can add 25/12 to both sides, giving

$$3\left(x + \frac{1}{6}\right)^2 = \frac{25}{12}. \tag{2.3.23}$$

Dividing by 3 we get

$$\left(x+\frac{1}{6}\right)^2=\frac{25}{36}. \tag{2.3.24}$$

To undo the squaring we take the square root, and we take ± as well, giving

$$x+\frac{1}{6}=\pm\frac{5}{6}. \tag{2.3.25}$$

Finally, we can subtract 1/6 from both sides giving the solution

$$x=-\frac{1}{6}\pm\frac{5}{6}, \tag{2.3.26}$$

which gives the solutions

$$x=-\frac{1}{6}-\frac{5}{6}=-1, \qquad \text{or} \qquad x=-\frac{1}{6}+\frac{5}{6}=\frac{2}{3}. \tag{2.3.27}$$

We can follow the same process as above but working with general $a$, $b$, and $c$. Starting with

$$a\left(x+\frac{b}{2a}\right)^2-\frac{b^2}{4a}+c=ax^2+bx+c, \tag{2.3.28}$$

we can add the constant term to each side,

$$a\left(x+\frac{b}{2a}\right)^2=\frac{b^2}{4a}-c. \tag{2.3.29}$$

Dividing by $a$ we get

$$\left(x+\frac{b}{2a}\right)^2=\frac{b^2}{4a^2}-\frac{c}{a}. \tag{2.3.30}$$

We can undo the squaring by taking square roots, remembering to include ± so we don't lose solutions:

$$x+\frac{b}{2a}=\pm\sqrt{\frac{b^2}{4a^2}-\frac{c}{a}}. \tag{2.3.31}$$

Some manipulation of fractions and square roots gives us

$$\sqrt{\frac{b^2}{4a^2}-\frac{c}{a}}=\sqrt{\frac{b^2-4ac}{4a^2}}=\frac{\sqrt{b^2-4ac}}{2a}. \tag{2.3.32}$$

Finally, adding $b/2a$ to both sides we end up with

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}, \tag{2.3.33}$$

which is exactly the quadratic equation!

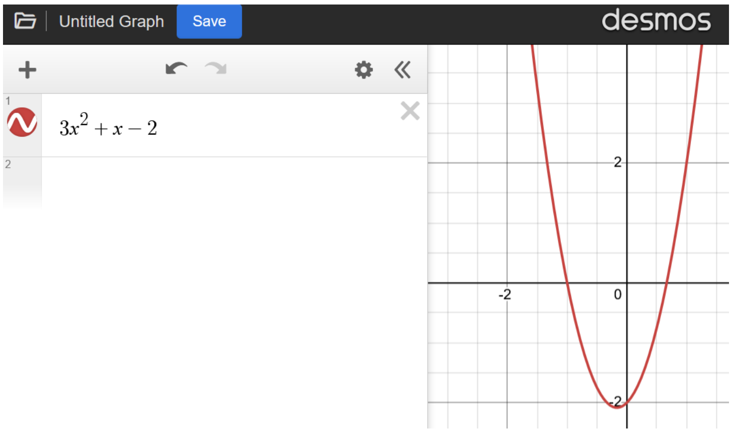


Figure 2.9: Plot of $y = 3x^2 + x - 2$ in Desmos.

### 2.3.4 Graphical Solution

If you can plot the quadratic then the solution is just where it crosses the $x$-axis. Figure 2.9 shows a plot done in Desmos. When you have this plot you can just hover the mouse over the line to find *approximate* values. This isn't a great method for finding solutions with one hundred percent certainty, but you can use it to guess solutions, $\alpha$ and $\beta$, then plug these into $(x - \alpha)(x - \beta)$ and expand, if you guessed correctly then you'll get the original quadratic back.

### 2.3.5 Computer Solution

The truth is that most people aren't solving quadratics manually. That being said it's important to understand quadratics as the second simplest (after a straight line) case of a polynomial. It's also a good way to learn about roots, turning points, and other properties of more general equations. This means that I can't, in good conscience, suggest that you just use a computer to solve all quadratics, but it can be done, and once you've had enough practice solving quadratics by hand it's a reasonable thing to do.

Note that a computer doesn't know if the solutions need to be real, so most will give you complex roots, which you can then choose to keep or exclude. If your solutions contain things like square roots of a negative, or the symbols $i$ or $j$ then that's a sign that the returned solution is complex.

**Code 2.3.34** Here's how to solve a quadratic equation in Matlab. This needs the "Symbolic Math Toolbox" add-on.

```
syms x;
solve(3*x^2 + x - 2 == 0)
>>> [-1, 2/3]
```

Here's how to solve a quadratic equation in Mathematica.

```
 In[1] Solve[3x^2 + x - 2 == 0]
Out[1] {{x -> -1}, {x -> 2/3}}
```

Here’s how to solve a quadratic equation in Python. This needs the “Sympy” package.

```
from sympy import solveset
from sympy.abc import x
solveset(3*x**2 + x - 2)
>>> {-1, 2/3}
```

### 2.3.6 Quadratic Inequalities

**Example 2.3.35** Solve

$$3x^2 + x - 2 \leq 0. \tag{2.3.36}$$

We already know that the two key points are $x = -2/3$ and $x = 1$. It just remains to see if the inequality is satisfied between these points our outside of them. Looking at Figure 2.9 we see that the graph dips below the $x$-axis, which is $y = 0$, between these points. So, we want between these points. Notice also that at these points $3x^2 + x - 2$ is 0, and we want to include 0 since we have $\leq$. Therefore, the solution set is the interval $[-2/3, 1]$.

# Three

# Binomial Theorem

## 3.1 Index Notation

Suppose you're designing a part for a car engine. One thing you may worry about is if the part can stand up to the temperatures in a car. You could start the car, wait a bit, then measure the temperature of the car. However, this isn't a great experiment, the car is likely to be a different temperature depending on how long its been running, how hard the engine is being pushed, the external temperature, and many other factors. A better experiment is to add a temperature logger to the car's engine which takes a temperature measurement, say every 5 minutes.

The nice thing about maths is that we can do some analysis of this data before we've even collected it! All we have to do is come up with a name for the data, then we can do maths with it without having to know the actual value! So, before you do this hypothetical experiment you might decide to give a name to every temperature you measure. A sensible choice is $T$ for temperature for the first measurement. Then the second measurement can be $t$. We can swap to Greek letters, calling the result of the next three measurements $\tau$, $\theta$ and $\Theta$. Now we have a problem, because we've run out of T-like letters, and soon we'll run out of all letters if we keep up like this. If we were taking hundreds of measurements this is a really bad way to label the results.

A much better idea is to call the first measurement $T_0$, then the second $T_1$ and the third $T_2$ and so on. We've chosen to start at 0, but you can start at 1 as well, you just have to adjust other things later. Similarly, we might label the times at which the measurements were made $t_0$, $t_1$, and so on. For short, we call our measurements $T_i$ and $t_i$, where $i$ is standing in for any **index** (that's what we call the subscript letter), which usually means $i = 1, \dots, N$, where $N+1$ is the total number of measurements we take.

One important value to consider in our experiment is the maximum temperature, which we might denote $\max\{T_i\}$. Another value that might be important, say if we're worried about thermal expansion, is the rate of temperature change. We can approximate the rate of temperature change between times $t_i$ and $t_{i+1}$ as

$$\frac{T_{i+1} - T_i}{t_{i+1} - t_i}. \tag{3.1.1}$$

Note that $t_{i+1} - t_i$ is just 5 minutes, so $(T_{i+1} - T_i)/5$ will give us the temperature change in degrees per minute (assuming $T_i$ is measured in degrees celsius).

Another important value is the average temperature, which can be computed as

$$\frac{T_0 + T_1 + \cdots + T_N}{N+1}. \tag{3.1.2}$$

This $\cdots$ is sometimes not quite as precise as we need. Instead, we have a special notation for sums like this.

**Notation 3.1.3 — Sigma Notation** If $T_0, \ldots, T_N$ are some values then we write

$$\sum_{k=0}^{N} T_k \tag{3.1.4}$$

for

$$T_0 + T_1 + \cdots + T_N. \tag{3.1.5}$$

**Remark 3.1.6** The $\sum$ symbol comes from the Greek letter $\Sigma$ (sigma), which is just a capital S for Sum.

We read Equation (3.1.4) as "the sum from $k = 0$ to $N$ of $T_k$". We call $k = 0$ and $N$ the **limits** of the sum. To evaluate the expression[1] $\sum_{k=0}^{N} T_k$ we start with $k = 0$, which gives us $T_0$, then $k = 1$, which gives $T_1$, and so on, up to $k = N$, which gives $T_N$. We sum up these results, giving $T_0 + T_1 + \cdots + T_N$.

**Example 3.1.7**

- $\displaystyle\sum_{k=0}^{5} T_k = T_0 + T_1 + T_2 + T_3 + T_4 + T_5$;
- $\displaystyle\sum_{r=2}^{7} r^2 = 2^2 + 3^2 + 4^2 + 6^2 + 7^2 = 114$;
- $\displaystyle\sum_{\ell=-5}^{-2} \ell(\ell+1) = -5(-5+1) + -4(-4+1) + -3(-3+1) + -2(-2+1) = 40$;
- $\displaystyle\sum_{i=4}^{6} \frac{2}{i}\alpha_i = \frac{2}{4}\alpha_4 + \frac{2}{5}\alpha_5 + \frac{2}{6}\alpha_6 = \frac{1}{2}\alpha_4 + \frac{2}{5}\alpha_5 + \frac{1}{3}\alpha_6$.
- $\displaystyle\sum_{i=1}^{3}\sum_{j=1}^{3} a_i b_j$, first evaluate the inner sum, $\sum_{j=1}^{3} b_j = b_1 + b_2 + b_3$, and then the outer one, giving

  $$a_1(b_1 + b_2 + b_3) + a_2(b_1 + b_2 + b_3) + a_3(b_1 + b_2 + b_3). \tag{3.1.8}$$

  Of course, you can expand this, and rearrange the terms to write it in different ways.

[1] Note that inline it's common to move the limits of the sum up and to the side.

**Problem 3.1.9** Evaluate the following sums[a]

- $\displaystyle\sum_{x=7}^{10}(x^2+2x);$
- $\displaystyle\sum_{\alpha=-2}^{2}|\alpha|;$
- $\displaystyle\sum_{s=1}^{10}s^3;$
- $\displaystyle\sum_{i=1}^{7}\sum_{j=1}^{2}(2i+3j)$

[a] Ans: 362, 6, 3025, 175

! Sometimes you'll see sums where one of the limits is $\infty$. You have to be careful with these sums because sometimes it doesn't make sense to sum together infinitely many things.

! Note that $i$ and $j$ are both commonly used as indices in sums. Don't confuse them with the complex unit, $\sqrt{-1}$.

**Remark 3.1.10** When you have to write lots of sums writing $k = 0$ and $N$ over and over gets pretty tiresome. You'll probably see people write

$$\sum_k T_k \qquad \text{or} \qquad \sum T_k. \tag{3.1.11}$$

You then have to work out where $k$ starts and finishes from context, or even that it's $k$ that is being summed in the first place! Avoid doing this until you are really comfortable with the sum notation, and avoid doing it in exams as well.

Using this notation we can write the average temperature as

$$\frac{1}{N+1}\sum_{k=0}^{N}T_k. \tag{3.1.12}$$

This notation is very useful, and very common, so make sure you're happy with it. There's a similar notation for multiplication.

**Notation 3.1.13** If $T_0, \dots, T_N$ are some values we write

$$\prod_{k=0}^{N}T_k \tag{3.1.14}$$

for

$$T_0T_1\cdots T_N. \tag{3.1.15}$$

**Remark 3.1.16** The $\prod$ symbol comes from the Greek letter $\Pi$ (pi) which is just capital P for product.

## 3.2 Factorials and Binomial Coefficients

**Definition 3.2.1 — Factorial** For $n \in \mathbb{N}$ we define the **factorial** of $n$, written $n!$ and read "$n$ factorial", to be the product of all of the whole numbers from 1 to $n$. That is,

$$n! = 1 \cdot 2 \cdot \cdots (n-1)n. \tag{3.2.2}$$

**Example 3.2.3** Here's $n!$ for $n = 1, \ldots, 10$:

| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $n!$ | 1 | 2 | 6 | 24 | 120 | 720 | 5040 | 40320 | 362880 | 3628800 |

Notice how quickly the factorial grows, by 10 we're already at about 3.6 million! [That's an exclamation mark, not another factorial]

If you start calculating some factorials by hand you'll realise that you're doing the same calculations over and over. For example, if we compute 6! we need to compute 5! along the way:

$$6! = \underbrace{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5}_{5!} \cdot 6. \tag{3.2.4}$$

In general, if we compute $n!$ we need to compute $(n-1)!$ along the way:

$$n! = \underbrace{1 \cdots (n-1)}_{(n-1)!}\, n. \tag{3.2.5}$$

This gives us a recursive way to compute factorials, which can also be taken as an alternative definition. For $n \in \mathbb{N}$ we have

$$n! := \begin{cases} 1 & n = 1; \\ (n-1)! \cdot n & n > 1. \end{cases} \tag{3.2.6}$$

**Code 3.2.7** Here's a python program which computes the factorial of an input:

```python
def factorial(n: int) -> int:
    if n == 1:
        return 1
    return n * factorial(n - 1)
```

Note that this is *not* an efficient way to compute this, but it's the most direct implementation of the equation above. Also, better code would check for invalid inputs, such as negative numbers, or non-integers, but that gets in

the way of understanding the maths.
Here's a similar implementation in Haskell

```haskell
factorial :: Int -> Int
factorial 1 = 1
factorial n = n * factorial (n - 1)
```

The first line just says the input and output are both integers. Haskell can automatically determine cases from this list of definitions.

One sensible question after we define something is "so what"? Why should we care about the factorial? One answer is that $n!$ is the number of ways to arrange $n$ (distinct) objects in a row. If we have the set $\{1, 2\}$ there are 2 different ways to order these elements, "12" or "21". If we have the set $\{1, 2, 3\}$ there are 6 different ways to order these elements, "123", "132", "213", "231", "312" or "321". We can keep going like this, there are 24 different ways to order the elements of $\{1, 2, 3, 4\}$, and so on.

Notice that we can rearrange the second part of the recursive definition to get that $(n-1)! = n!/n$. Using this we can extend the definition of factorial to 0. If $n - 1 = 0$ then $n = 1$ and we have

$$0! = \frac{1!}{1} = \frac{1}{1} = 1. \tag{3.2.8}$$

We then extend our definition of factorial to all non-negative integers by defining $0! \coloneqq 1$.

**Remark 3.2.9** There are several other justifications for why $0! = 1$ is the correct definition to make. At the end of the day we make definitions in maths because they're useful, and $0! = 1$ is the most useful definition, mostly because it fits into lots of patterns, including the following.

- Think about $n!$ as the number of ways of arranging $n$ things. If we have $\{1\}$ then there's only one way to order the elements, "1". If we have $\varnothing = \{\}$ then there's also only one way to order the elements, "" (that's the empty list, but it's still a valid ordering of no things).

- Next year you'll learn some complex analysis, in that course you may come across the gamma function,

$$\Gamma(z) \coloneqq \int_0^\infty t^{z-1} \mathrm{e}^{-t} \, \mathrm{d}t. \tag{3.2.10}$$

  Don't worry about any of these symbols if they're not familiar or just a bit scary. The important thing is that you can show that the gamma function is such that $\Gamma(n) = (n-1)!$ for any $n \in \mathbb{N}$, and also $\Gamma(1) = 1$ so we should define $(1-1)! = 0! = 1$ if we want this pattern to continue. The gamma function actually allows us to extend the factorial to negative numbers and even non-whole numbers! For example, you can evaluate this function at $3/2$ and you find that $\Gamma(3/2) = \sqrt{\pi}$. If you spend enough time in proximity to "Popular Maths", say on

> Youtube, you'll probably see someone claiming that 1/2 factorial is $\sqrt{\pi}$, and this is what they mean. It isn't really correct to call this the factorial though unless the input is a positive whole number (which then includes $0! = \Gamma(1)$ since 1 is positive).

A reasonable question, following on from the question of ordering things, is the following. If I have a set of $n$ objects how many different ways can I pick $k$ of them? When faced with a question like this it's often good to look at a few examples with small numbers. If $k = 0$ then there's one way to pick a set of 0 elements from $\{1, \dots, n\}$, you take $\varnothing$. Similarly, if $k = n$ then there's only one way to pick a set of $n$ elements from $\{1, \dots, n\}$, you take $\{1, \dots, n\}$. If $k = 1$ then there's $n$ ways to pick a set of 1 element from $\{1, \dots, n\}$, we could take $\{1\}$, $\{2\}$, and so on up to $\{n\}$.

If $k = 2$ then things are a bit trickier, so let's fix a value of $n$ as well, say $n = 4$. We can take $\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}$, or $\{3,4\}$. That's 6 different ways to make the choice. If you keep playing around with small values, and if you pick the sets in a sensible order, then you may start to see a pattern emerging. The key is to think about ordering the items, and then forgetting the orders again at the end. We've been doing this implicitly by choosing to call our elements $1, \dots, n$.

Here's a way to pick $k$ elements from any $n$ element set. First, order the $n$ elements in some way. There are $n!$ ways to do this. Then pick the first $k$ elements. There's a problem with this though, if I swap two elements amongst the first $k$ the result doesn't change, and if I swap two elements among the last $n - k$ the result doesn't change again. For example, if $n = 6$ and $k = 3$ then I can order my elements as 123456. Then I take the first three elements, 123. Forgetting the order this gives us the set $\{1, 2, 3\}$. However, if I'd picked any of the orders 132456, 123465, 132465, and so on I would still result in picking $\{1, 2, 3\}$ as the final set (although the order may be different, but that's not important for a set).

What we see is that the $n!$ ways to order these elements results in us over counting the number of subsets of size $k$. In particular, we over count by a factor of the number of ways of rearranging the first $k$ elements, which is $k!$, and a factor of the number of ways of arranging the last $n - k$ elements, which is $(n - k)!$. The result is that the number of ways of picking $k$ things from a set of $n$ things is

$$\frac{n!}{k!(n-k)!}. \tag{3.2.11}$$

> **Definition 3.2.12 — Binomial Coefficient** The **binomial coefficient** is defined by
>
> $$\binom{n}{k} = {}_nC_k = {}^nC_k := \frac{n!}{k!(n-k)!}. \tag{3.2.13}$$

The reason for the name "binomial coefficient" will become clear in the next section.

> **Problem 3.2.14** Compute the following binomial coefficients[a]. For the first three try using the formula and counting the number of subsets. For the

others you can just use the formula

$$\binom{5}{3};\quad \binom{6}{3};\quad \binom{4}{2};\quad \binom{10}{9};\quad \text{and}\quad \binom{10}{5}. \tag{3.2.15}$$

Hint: Finding subsets with 9 out of 10 things is the same as finding subsets which don't contain 1 out of 10 things.

[a] Ans: 10, 20, 6, 10, 252

### 3.2.1 Binomial Expansion

A **binomial** is any expression of the form $a+b$. The expansion part is that we often need to compute $(a+b)^2$, $(a+b)^3$, or even higher powers. Fortunately, there's a fast way to do it. We can use the **binomial theorem**.

**Theorem 3.2.16 — Binomial Theorem.** For $n \in \mathbb{N}$ we have

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k. \tag{3.2.17}$$

**Example 3.2.18** Compute $(x+3)^2$:

$$\begin{aligned}
(x+3)^2 &= \sum_{k=0}^{2} \binom{2}{k} x^{2-k} 3^k && (3.2.19)\\
&= \binom{2}{0} x^{2-0} 3^0 + \binom{2}{1} x^{2-1} 3^1 + \binom{2}{2} x^{2-2} 3^2 && (3.2.20)\\
&= x^2 + 2x \cdot 3 + 9 && (3.2.21)\\
&= x^2 + 6x + 9. && (3.2.22)
\end{aligned}$$

Compute $(a+b)^3$:

$$\begin{aligned}
(a+b)^3 &= \sum_{k=0}^{3} \binom{3}{k} a^{3-k} b^k && (3.2.23)\\
&= \binom{3}{0} a^{3-0} b^0 + \binom{3}{1} a^{3-1} b^1 + \binom{3}{2} a^{3-2} b^2 + \binom{3}{3} a^{3-3} b^3 && (3.2.24)\\
&= a^3 + 3a^2 b + 3ab^2 + b^3. && (3.2.25)
\end{aligned}$$

Compute $(x+y)^4$:

$$\begin{aligned}
(a+b)^4 &= \sum_{k=0}^{4} \binom{4}{k} x^{4-k} y^k && (3.2.26)\\
&= \binom{4}{0} x^4 y^0 + \binom{4}{1} x^3 y^1 + \binom{4}{2} x^2 y^2 + \binom{4}{3} x^1 y^3 + \binom{4}{4} x^0 y^4 && (3.2.27)
\end{aligned}$$

$$= x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4. \tag{3.2.28}$$

Some things to notice which help you avoid errors:

- If you add up the exponents (including 1) in each term you'll always get $n$. In the last example we get $4+0$, $3+1$, $2+2$, $1+3$, and $0+4$, all of which add up to 4.
- The coefficients should increase up to some value, then decrease in reverse. For these three examples the coefficients go $1, 2, 1$, then $1, 3, 3, 1$, then $1, 4, 6, 4, 1$.

The nice thing about the binomial theorem (also known as the binomial formula) is that you don't always have to compute the whole thing. Suppose you just want to know what the coefficient of $h^{32}$ is in $(x/y + h)^{50}$. You can compute this, it's just

$$\binom{50}{32}\left(\frac{x}{y}\right)^{50-32} = 18053528883775\frac{x^{18}}{y^{18}}. \tag{3.2.29}$$

Okay, that's a bit of a silly example, but it really is useful to be able to quickly determine coefficients of single terms sometimes.

**Example 3.2.30** Suppose you're computing interest. If the interest is paid at a rate of 5% annually then after 3 years the amount is $1.05^3N$ where $N$ is the original amount. If you don't have a calculator you can compute $1.05^n$ using the binomial expansion. First, write $1.05 = 1 + 0.05$ and note that $0.05 = 5/100$. Then we have

$$\begin{aligned}
1.05^3 &= \left(1 + \frac{5}{100}\right)^3 && (3.2.31)\\
&= \sum_{k=0}^{3}\binom{3}{k}1^{3-k}\frac{5^k}{100^k} && (3.2.32)\\
&= \binom{3}{0} + \binom{3}{1}\frac{5}{100} + \binom{3}{2}\frac{5^2}{100^2} + \binom{3}{3}\frac{5^3}{100^3} && (3.2.33)\\
&= 1 + 3\cdot\frac{5}{100} + 3\cdot\frac{25}{10000} + \frac{125}{1000000} && (3.2.34)\\
&= 1 + \frac{15}{100} + \frac{75}{10000} + \frac{125}{1000000} && (3.2.35)
\end{aligned}$$

If we only care about computing things to the nearest thousandth then we can approximate this as

$$1.05^3 \approx 1 + \frac{15}{100} = 1 + 0.15 = 1.15. \tag{3.2.36}$$

If we want more accuracy we can include more terms. Including all of them gives

$$1 + 0.15 + 0.0075 + 0.000125 = 1.157625. \tag{3.2.37}$$

All of that can be done without a calculator, and actually gave one more decimal place of accuracy than the first calculator I checked it with!

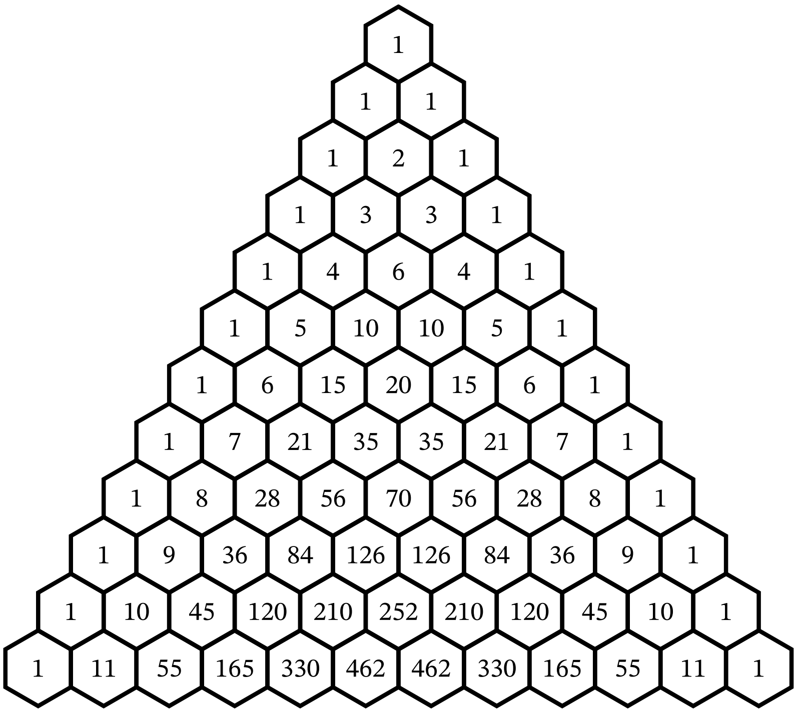


Figure 3.1: Pascal's triangle.

> **Remark 3.2.38** Computing approximations like this, where we are adding smaller and smaller terms, will generalise to the notion of Taylor series. We'll discuss these in Block 4.

As you practice this you'll likely learn the coefficients for the first few exponents off by heart. In fact, there's a chance you recognise them already. Pascal's triangle (Figure 3.1) is constructed by starting with 1s on two sides. Then the rule to fill in the triangle is that each element is the sum of the two elements above it to either side. The result is that the rows are exactly the coefficients of the binomial expansion corresponding to that row number, that is, the binomial coefficients!

# Part II

# Geometry

# Four

# Coordinates and Conics

## 4.1 Coordinates

Where are you? How would you tell someone your exact location right now? For me, I'm in my office, at my desk. It's the second desk on the right. Can I be more accurate than that? I'm about 3 m from the wall with the door, and 1 m from the wall to the right of that.

However, that's only useful if you already know where my office is. Maybe a more useful position would be my latitude, 55.872 480, and longitude, −4.294 590. These two numbers can pinpoint any place on the surface of Earth. Latitude is measured as an angle above or below the equator, and longitude as an angle East or West of the Prime Meridian (an arbitrary line drawn from pole-to-pole, originally chosen to pass through Greenwhich observatory, now slightly off from this).

What I've just done is give you some coordinates to position me in my office. **Coordinates** are numbers which specify a position relative to something else. In the first case relative to the walls of my office, and in the case of latitude and longitude relative to the equator and IERS Reference Meridian.

There are many more choices I could have made to specify my location, say my grid position in an Ordnance Survey map. These coordinates are all useful in the real world. We'll look at some more idealised coordinates which are useful for solving both real-world and mathematical problems.

> **Remark 4.1.1** The study of coordinates generalises to the definition and study of https://en.wikipedia.org/wiki/Manifold, which intuitively are any objects where we can use coordinates to specify locations. However, the notion of coordinates for manifolds is much more flexible than we have here. We're working with Euclidean spaces, which come with a fixed notion of distance. Manifolds don't have this restriction.

### 4.1.1 Cartesian Coordinates

Hopefully, you're familiar with coordinates in the plane. To specify a point in the plane you can fix two orthogonal axes, $x$ and $y$, and specify any position by how far along each axis you have to go. We call these **Cartesian coordinates**, named after René Descartes. See Figure 4.1.

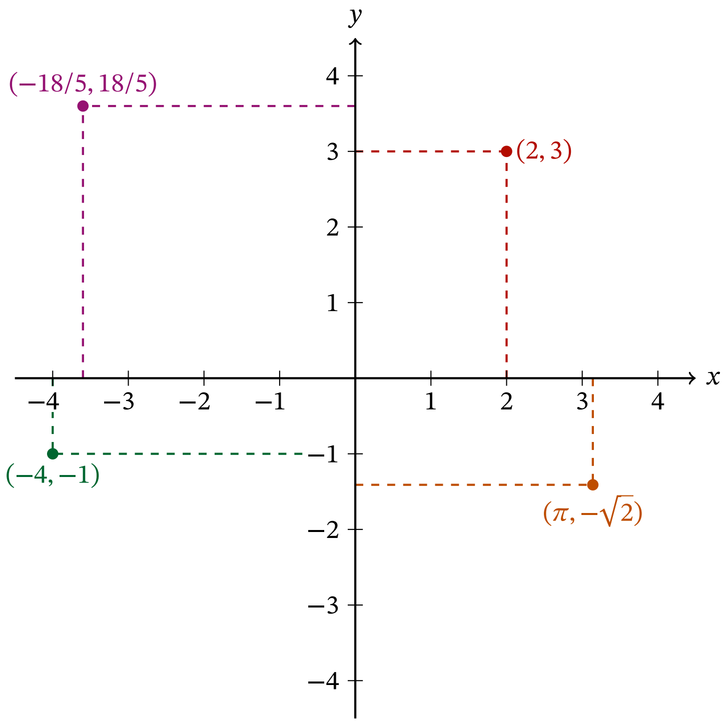


Figure 4.1: Cartesian Coordinates on the Plane

Once we've fixed the axes we call the point they cross the **origin**. Any position in the plane can then be given as a pair of numbers, $(x, y)$, giving the distance along the $x$ and $y$ axis respectively. The origin has coordinates $(0, 0)$.

**Notation 4.1.2 — Tuples** Given a set, $S$, we write $S^n$ to mean the set of all $n$-tuples of $S$. That is,

$$S^n = \{(s_1, \dots, s_n) \mid s_i \in S\}. \tag{4.1.3}$$

With this notation the Cartesian coordinates of a point on the plane belong to $\mathbb{R}^2$, since in theory any real number can appear as a length (although in practice we can only measure rational numbers).

Cartesian coordinates work in any number of dimensions. In fact, the number line that we've mentioned several times already is really just the Cartesian coordinates of the line.

You need as many coordinates as you have dimensions. In fact, this is pretty much the definition of **dimension**, it is the number of (independent) pieces of information that you need to specify to locate a point. So, in three dimensions you need three pieces of information.

In Cartesian coordinates the third dimension means adding a third axis, $z$. Then our coordinates live in $\mathbb{R}^3$. An example is given in Figure 4.2.

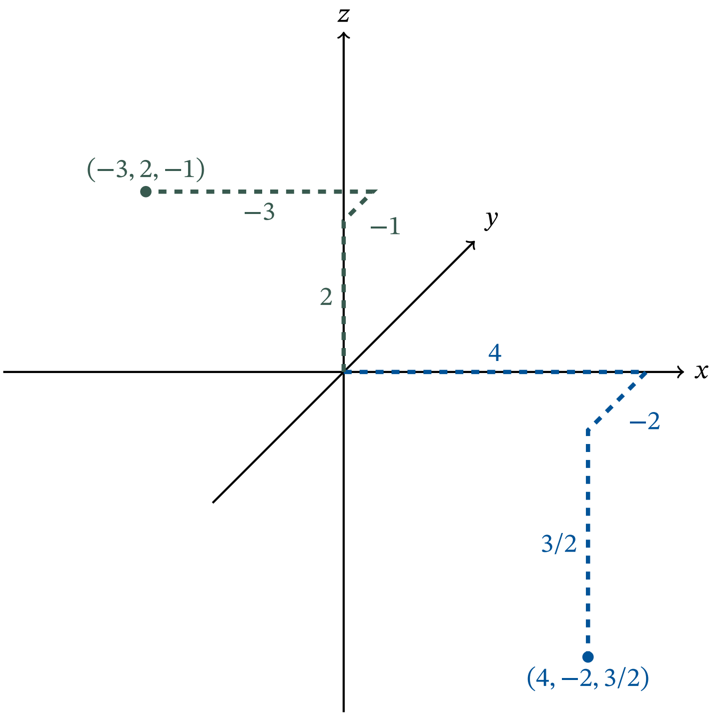


Figure 4.2: Cartesian coordinates in three dimensions. To find the coordinates of a point imagine you start at the origin, then move to the point, but you're only allowed to move parallel to the axes. Keep track of how far you go along each axis. Notice that you don't need to start with the $x$-axis. In fact, you can walk along the $x$-axis for a bit, then the $y$-axis, and then the $x$-axis again. As long as you keep track of each axis independently you'll always end up with the same coordinates once you reach the point.

**Remark 4.1.4** We will always choose to label our axes in accordance with the right hand rule, this will be important when you learn about the vector cross product. It's also a convention that is followed pretty much universally.

It's possible to keep adding axes, however since we live in three dimensional space it becomes hard to plot. Rather than trying to picture what the fourth dimension looks like I find it easier to switch up my mental picture of what coordinates are. They're just separate pieces of data which come together to specify some combined piece of information. For example, if you have a beam then the stresses in the beam are measured by 9 numbers, which can be taken as the stress in each direction along three perpendicular faces of the beam. Thus, this information lives in a 9 dimensional space.

**Remark 4.1.5** This information about the stresses in a beam is usually packaged up into the Cauchy stress tensor. In many practical situations

> these different numbers aren't actually independent, and we can reduce the amount of information needed. For example, in equilibrium only 6 numbers are needed.

#### 4.1.1.1 Geometry in Cartesian Coordinates

Let's return to the case of the plane again. You should be familiar with the equation[1]

$$y = mx + c. \tag{4.1.6}$$

Once we fix values for $m$ and $c$ plotting this will give a straight line with gradient $m$ and $y$-intercept $c$.

Recall that the gradient is a measure of how steep the line is. To measure the gradient find two points, $(x_1, y_1)$ and $(x_2, y_2)$, on the line. Then the **gradient** is the change in $y$-value (known as the **rise**) divided by the change in the $x$-value (known as the **run**):

$$m = \frac{\Delta y}{\Delta x} = \frac{y_1 - y_2}{x_1 - x_2}. \tag{4.1.7}$$

Note that a positive gradient means the line slops up when coming from the right, and a negative gradient means it slopes down. The larger the absolute value of $m$ is the steeper the slope. A horizontal line has a gradient of 0, and a vertical line has, in a sense, an infinite gradient. More properly, it has an *undefined* gradient.

> **Remark 4.1.8** In block 4 we'll see how this generalises to any curve, defining the gradient more generally using the derivative.

The ***y*-intercept** is simply the $y$-value when the line passes through the $y$-axis, which is $x = 0$, and when $x = 0$ we have $y = m \cdot 0 + c = c$.

> **Example 4.1.9** Consider Figure 4.3. Here the line plotted passes through the $y$-axis at $x = 1$, so the $y$-intercept is $c = 1$. Making a measurement we see that when we go along by 2.5 we have to go up by 5 to get back to the line. Thus, the gradient is
>
> $$m = \frac{5}{2.5} = 2. \tag{4.1.10}$$
>
> Hence, the equation of this line is
>
> $$y = 2x + 1. \tag{4.1.11}$$
>
> You can check this by finding the $x$ and $y$ coordinates of any point on the line and checking that they satisfy this formula.

You may be familiar with equations of the form

$$(y - y_0)^2 + (x - x_0)^2 = r^2. \tag{4.1.12}$$

[1] The conventions used in this equation differ from country to country. I've chosen the convention most commonly used in the UK. Other common conventions are $y = mx + b$ and $y = ax + b$.

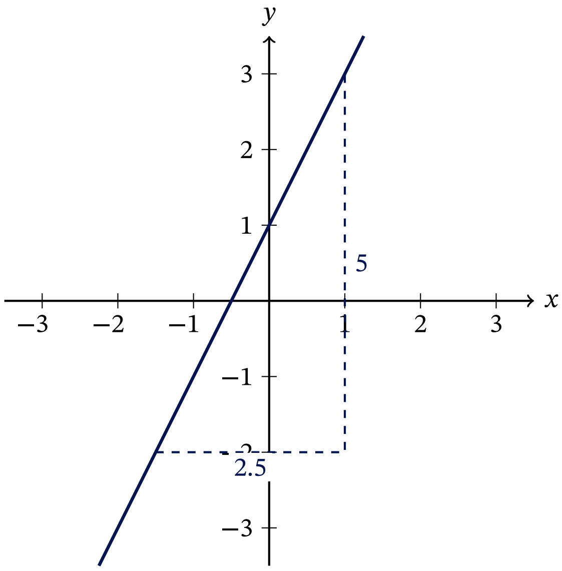


Figure 4.3: Straight line of gradient 2 and $y$-intercept 1. The equation of this line is $y = 2x + 1$.

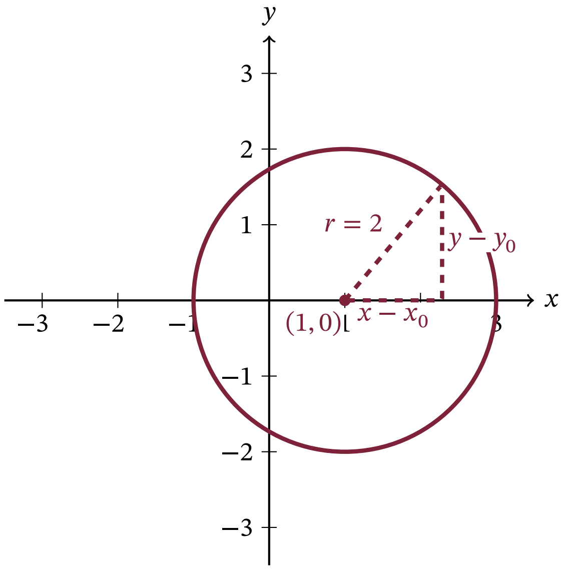


Figure 4.4: Circle centre $(1, 0)$ and radius 2. The equation of this circle is $(x-1)^2 + y^2 = 4$.

Once we fix values for $x_0$, $y_0$, and $r$ this will give a circle with centre $(x_0, y_0)$ and radius $r$. The reason for this is simply Pythagoras. Notice that $x - x_0$ and $y - y_0$ are the distances along each axis from the centre of the circle, and so we get the triangle shown in Figure 4.4, which has the radius of the triangle as its hypotenuse.

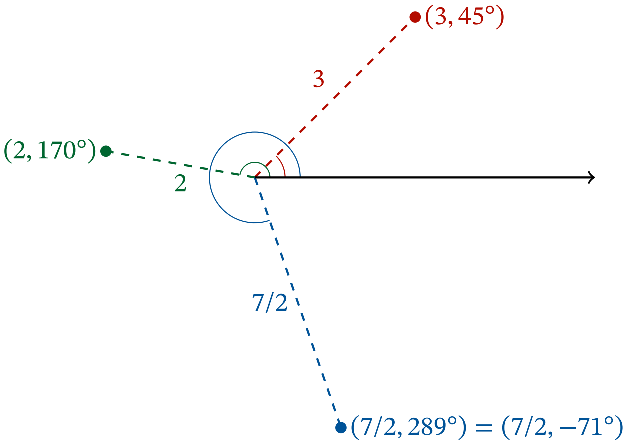


Figure 4.5: Polar coordinates. All angles are measured from the horizontal line going anticlockwise. Notice that $-71 = 289 - 360$. Negative angles are measured clockwise.

### 4.1.2 Polar Coordinates

While Cartesian coordinates can be used to describe any point in, say, the plane they aren't always the best choice. In particular, if the problem at hand has some degree of rotational symmetry about the origin it is often better to use coordinates which reflect this. For this reason we now introduce polar coordinates.

To specify a point in the plane in polar coordinates we start with a choice of origin and a straight line or **ray** starting at the origin going in some chosen direction, which usually we take to be horizontally to the right. To specify a point we give two pieces of information:

- The distance, $r$, from the origin.
- The angle, $\theta$, of a line from the origin to the point, measured from the chosen line in the anticlockwise direction (or negative if measured clockwise).

Then the coordinates of the point are $(r, \theta)$. See Figure 4.5 We have to make a convention choice about the range of values $\theta$ can take. One common choice is $\theta \in (0°, 360°)$. Another common choice is $\theta \in (-180°, 180°)$. Either is fine, but it's important to restrict the range if we want unique coordinates. If we don't mind about unique coordinates then $(r, \theta)$ and $(r, \theta + n \cdot 360°)$ both describe the same point for any $n \in \mathbb{Z}$.

! If you just write $(a, b)$ it's ambiguous what you mean. Is this point in Cartesian coordinates or polar coordinates? Often it's clear from context, but you should always specify.

! Conventions for what we call polar coordinates aren't as fixed as they are for Cartesian coordinates. You might see $\rho$ in place of $r$ and $\varphi$ in place of $\theta$. You may also see $(\theta, r)$ instead of $(r, \theta)$, so always check the conventions of any source you use.

#### 4.1.2.1 Geometry in Polar Coordinates

A straight line through the origin has a particularly simple equation in polar coordinates, it's just

$$\theta = \alpha \text{ or } \theta = \alpha + \angle 180. \tag{4.1.13}$$

Picking all points at a constant angle, $\alpha$, gives a ray going from the origin, and picking all points at the angle $\alpha + \angle 180$ gives the other half of the line.

The equation form is simple (despite having to split into two cases) because we have the symmetry that rotating around the origin doesn't change the fact that the line passes through the origin. In general we want to use polar coordinates where there's some sort of symmetry when we rotate around the origin.

Another example is a circle centred on the origin, which doesn't change at all when we rotate around the origin. This results in an even simpler equation for a circle centred at the origin in polar coordinates. It's simply

$$r = R \tag{4.1.14}$$

for a circle or radius $R$. A circle is, by definition, the set of all points a fixed distance from the origin, and in polar coordinates that's specified by fixing $r$ and letting $\theta$ vary around the circle.

## 4.2 Conic Sections

So far we've seen

- straight lines;
- circles;
- parabolas.

It turns out that these, plus a few other types of curves, can all be understood as **conic section**. These are curves that arise when we take a conic (two cones tip-to-tip) like in **??** and look at how it intersects a plane.

Figure 4.7 shows the curves that we can get this way. The full list is as follows:

- point;
- straight line;
- two intersecting straight lines;
- circles;
- ellipses;
- hyperbolas;
- parabolas.

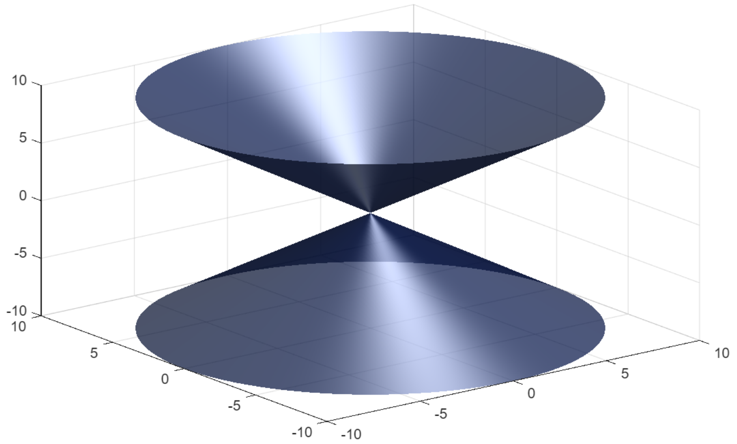


Figure 4.6: Two cones, tip-to-tip, as required to define conic sections.

Let's start at the top. There's only one way to get a single point, you have to have the plane pass through the tip of the cones and nothing else.

For a straight line have the plane at the same angle as the sides of the cone, and line it up so it *just* touches the cones and goes through the tip of the cones.

For two intersecting straight lines take a vertical plane through the tip of the cones.

For a circle take a horizontal plane through the cones.

For an ellipse take a plane on an angle between horizontal and the angle of the cone walls. An ellipse is just a squashed circle. An ellipse centred on the origin has the equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1. \tag{4.2.1}$$

Here $a$ and $b$ are the distances from the origin to the ellipse along the $x$ and $y$ axis. Notice that if $a = b = r$ we can multiply through by $r^2$ and we get the equation of a circle centred on the origin. A circle is just a special case of an ellipse.

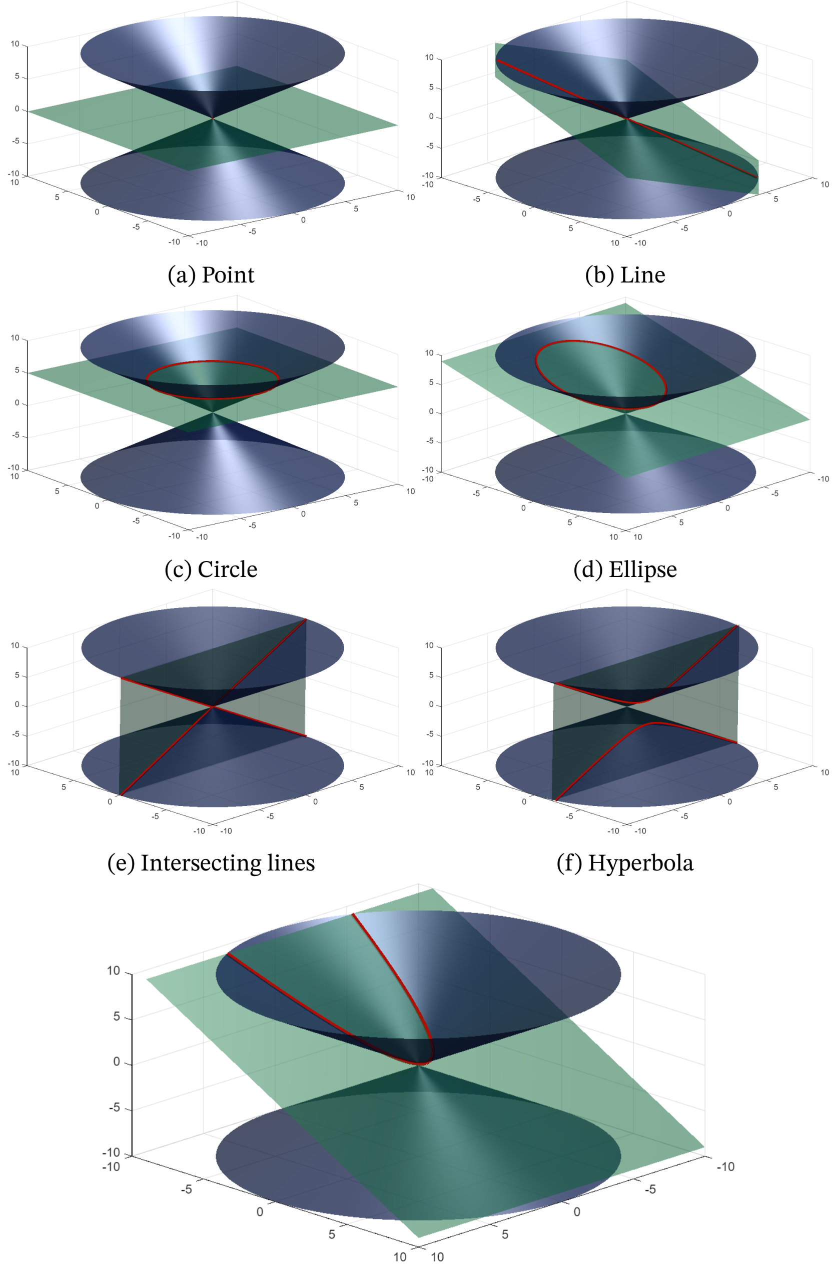
For a hyperbola take the plane to be at some angle greater than the angle of the sides of the cone and less than vertical. The intersecting straight lines are the special case where the plane is vertical and passes through the tip of the cones. Note that there are two disconnected lines making up the hyperbola, but it's still considered to be one single curve.

Finally, for a parabola take the plane on the same angle as the sides of the cone, but not through the tip.

**Code 4.2.2** Here's the code used to create these diagrams. This is modified from code from Sage Stanish, a previous lecturer.

```
a = 0;
b = 1;
c = 1;

[x, y] = deal(linspace(-10, 10, 100));
```



(a) Point

(b) Line

(c) Circle

(d) Ellipse

(e) Intersecting lines

(f) Hyperbola

(g) Parabola

Figure 4.7: Conic sections

```
[X, Y] = meshgrid(x, y);

p_cone = sqrt(X.^2 + Y.^2);
plane = c - a*X - b*Y;

p_diff = p_cone - plane;
C = contours(X, Y, p_diff, [0, 0]);
xL = C(1, 2:end);
yL = C(2, 2:end);
zL = interp2(X, Y, p_cone, xL, yL);
zL2 = interp2(X, Y, -p_cone, xL, yL);

clf;
surf(X, Y, p_cone, ...
    "FaceColor", blue, "FaceAlpha", 0.7, ...
    "EdgeColor", "none");
hold on;
surf(X, Y, -p_cone, ...
    "FaceColor", blue, "FaceAlpha", 0.7, ...
    "EdgeColor", "none");
surf(X, Y, plane, ...
    "FaceColor", green, "FaceAlpha", 0.5, ...
    "EdgeColor", "none");
line(xL, yL, zL, "Color", red, "LineWidth", 3);
% For a point the intersection isn't computed
% properly, so need to add point manually
% line(0, 0, 0, ...
%   "Marker", ".", "Color", red, "LineWidth", 3);
% Same as above but for a line
% line([7, -7], [7, -7], [-10, 10], ...
%   "Color", red, "LineWidth", 3);
view([1, 1, 1])
camlight;
axis([-10, 10, -10, 10, 0, 10]);
hold off;
```

Try the following values of $(a, b, c)$:

$$(0, 0, 0), (0, 0, 5), (0, 1, 1), (0, 0.5, 1),$$
$$(0, 100, -200), (0, 100, 0), (0.7, 0.7.0).$$

They should give the point, circle, parabola, ellipse, hyperbola, intersecting lines, and single line respectively.

The formula for a hyperbola is

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1. \tag{4.2.3}$$

It's just like the formula of an ellipse but with $-$ instead of $+$.

**Application 4.2.4** Consider a jet flying faster than the speed of sound. The speed of sound is closely related to how fast air can respond to movement. When the jet exceeds the speed of sound the air can't get out of the way fast enough, and it gets compressed. This compressed air then sheds off the jet, moving outwards, leaving a cone of high pressure air behind the jet. It is this high pressure air which is hear as a sonic boom.

Since this high pressure air forms a cone the intersection of this cone and the ground (which is essentially flat on the scale in question) is a conic section. Assuming that the jet is flying parallel to the ground and above ground level it will be (half of) a hyperbola.

**Remark 4.2.5** You may be familiar with the famous equation

$$E = mc^2. \tag{4.2.6}$$

It relates the energy of an object to its mass. However, this is only true if the object is stationary. If the object is moving instead then it has nonzero momentum, $p$, and the correct formula for the total energy (kinetic plus the mass-energy equivalence) is

$$E^2 = m^2c^4 + p^2c^2. \tag{4.2.7}$$

A particle on its own has a fixed energy. Rearranging this equation we get

$$\frac{p^2}{m^2c^2} - \frac{E^2}{m^2c^4} = 1, \tag{4.2.8}$$

which relates $x = p$ and $y = E$ as plotting a hyperbola. In particle physics this hyperbola is known as the mass-shell, and all (real) particles must have their momentum, energy and mass balanced so that they appear somewhere on this hyperbola. Note that in reality momentum actually has components in all three directions, so we get a hyperbeloid.

Any conic section has an equation which can be written in the form

$$ax^2 + by^2 + 2fx + 2gy + 2hxy + c = 0 \tag{4.2.9}$$

for some values $a$, $b$, $c$, $f$, $g$, and $h$. In particular, ignoring the edge-cases of a point or straight lines we get

- a circle if $a = b \neq 0$ and $h = 0$;
- a parabola if $h^2 = ab$;
- an ellipse if $h^2 < ab$;
- and a hyperbola if $h^2 > ab$.

**Problem 4.2.10** Here's a desmos[a] implementation of this equation. Try to pick values for $a$, $b$, $c$, $f$, $g$, and $h$ which give you each case of a conic section.

[a] https://www.desmos.com/calculator/0vtj4tzax8

If we're content to have our conics in some sense pinned to the origin we can express the equation in polar coordinates as

$$r = \frac{\ell}{1 + e\cos\theta} \tag{4.2.11}$$

where $\ell$ (which is just a curly $l$) and $e$ are parameters. This is a particularly nice form because the parameters $\ell$ and $e$ tell us quite a lot. The value of $\ell$ is just a measure of the "size" of the conic section, for a circle it's just the radius. We call $e$ the **eccentricity**. For a circle $e = 0$, for a parabola $e = 1$. For $e \in [0, 1)$ we have an ellipse, and for $e > 1$ we have a parabola. This value $e$ is related to the angle of the plane, with $e = 0$ being the horizontal plane, and $e = 1$ being the plane parallel to the edges of the cone.

**Application 4.2.12 — Orbital Mechanics** Gravity provides an attractive force. To a very good approximation this force is given by Newton's law of gravity, which states that the force on an object of mass $m$ due to an object of mass $M$ at a distance $r$ is given by

$$F = \frac{GMm}{r^2}. \tag{4.2.13}$$

Here $G = 6.67 \times 10^{-11}\ \mathrm{m}^3\,\mathrm{kg}^{-1}\,\mathrm{s}^{-2}$ is known as the gravitational constant. It turns out that when you have an inverse square law like this, that is, when the force is proportional to $1/r^2$, the only orbits you can have are conic sections (assuming there are only two objects). For example, the Earth orbits the sun in an ellipse, although it's pretty close to a perfect circle, with an eccentricity of $e = 0.016$.

There are some comets which orbit the sun, but at such a large distance that they enter and leave the solar system. Most famously, Halley's comet has an eccentricity of $e = 0.967$, and passes Earth approximately once every 80 years (We'll next see it some time in 2061).

The orbits of Earth and Halley's comet are both bound, meaning that over enough time the average distance from the Sun isn't changing. This isn't quite true, because the presence of other bodies, like Jupiter, move us away from the ideal situation of two body orbits, but three body orbits are famously a hard problem (and excellent book). However, the effects of these things are often small enough that if we want to correct for them then we also need to use general relativity instead of Newtonian gravity, so we'll leave those problems alone.

There are also unbound orbits. These aren't orbits in the typical sense of going round and round. Instead, they're orbits in the sense that they are objects following paths dictated only by the gravitational force from some much more massive central object (such as the Sun). Unbound orbits include those of things like comets which pass through the solar sys-

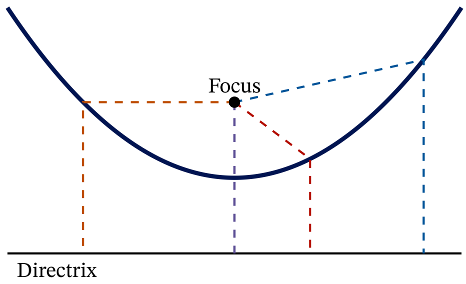


Figure 4.8: A parabola can be defined in terms of its focus and directrix. All dashed lines are split into two segments of equal length.

> tem never to return again. They will mostly do so on a hyperbolic path (or a parabolic one, although this is just as unlikely as a perfectly circular orbit).

Here's a method for constructing a parabola without making any measurements. Draw a straight line, $L$. Fix a point, $P$, not on the line. The parabola is the shape given by all points which are the same distance from both the point and the line. The line is called the **directrix** and $P$ is called the **focus**. See Figure 4.8.

> **Application 4.2.14** If you take a parabola and rotate it around its axis of symmetry the shape that you get is a **paraboloid**. These have the nice property that if you place a light source at the focus of a mirrored paraboloid then any light which bounces off the paraboloid will leave the paraboloid in parallel. This is useful if you're designing, for example, a torch. You can place the bulb at the focus of the paraboloid and then the light leaving the torch will form a nice beam.

Here's a method for constructing an ellipse without making any measurements. Fix two points, $P_1$ and $P_2$. Take a piece of string and attach one end to each point. Pull the string tight in the middle, and mark a point where the string folds. Do the same pulling the string tight part way along and marking where it folds. Keep doing this, and the points you mark will all be on the same ellipse. Another way to phrase this is that an ellipse consists of all points where the total distance from that point to $P_1$ plus the total distance from that point to $P_2$ is constant (the length of the string). Note that if we take $P_1$ and $P_2$ to be the same point then we just get a circle with radius half the length of the string. We call $P_1$ and $P_2$ the **foci**[2] of the ellipse. See Figure 4.9.

> **Application 4.2.15** If you have a mirrored ellipse then any light source placed at one focus will bounce off the ellipse and hit the other focus. This makes for a very boring pool table, so long as you start at one focus and the pocket is placed at the other focus (which is admittedly not at the edge of the table) you can't miss!

[2] the plural focuses is also acceptable

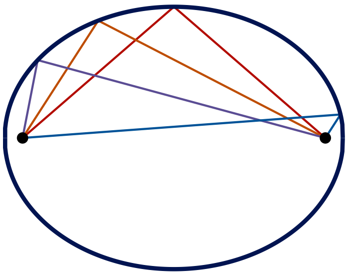


Figure 4.9: An ellipse can be defined in terms of its two foci. The total length of any line here is the same.

# Five

# Numbers and Errors

In pure maths we can get away with assuming all of our numbers are known to infinite precision, and the way we represent them doesn't matter. That doesn't really match reality though. In this chapter we'll look at different ways of representing numbers and how things like rounding errors accumulate through a calculation.

It's important to accept that even with the best computers and measuring devices any measurement we make, and any computation we do with it, is only able to approximate the "true" value. So we can't avoid rounding errors.

## 5.1 Bases

You're familiar with base 10, it's the way we normally represent numbers. You may also have heard of binary, usually in relation to computers. If you've ever seen a hex code for a colour, (for example, this red is `#B30C00`), then you've seen hexadecimal. These are all ways of representing numbers.

In base 10 if we have a number like 123 then what this really means is we have 1 lot of 100, 2 lots of 10, and 3 lots of 1. In other words,

$$123 = 1 \cdot 100 + 2 \cdot 10 + 3 \cdot 1. \tag{5.1.1}$$

Now notice that $100 = 10^2$, $10 = 10^1$, and $1 = 10^0$. Thus,

$$123 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0. \tag{5.1.2}$$

We can extend this, if we have 123.45 then this is

$$123.45 = 123 + 4 \cdot 0.1 + 5 \cdot 0.01 \tag{5.1.3}$$

and $0.1 = 10^{-1}$ and $0.01 = 10^{-2}$, so

$$123.45 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 + 4 \cdot 10^{-1} + 5 \cdot 10^{-2}. \tag{5.1.4}$$

In general, if we write $a_i$ for each digit we have

$$a_n a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m} = \sum_{k=-m}^{n} a_k \cdot 10^k. \tag{5.1.5}$$

A sensible question is then why we choose 10. Most arguments for why come down to the simple fact that we (well, most of us anyway) have ten fingers. There are other sensible choices that we could make. The choice of ten is called **decimal**. The number we choose is called the **base**.

A general formula for expressing a number in base $d$ is then

$$(a_n a_{n-1} \ldots a_1 a_0 . a_{-1} a_{-2} \ldots a_{-m})_d = \sum_{k=-m}^{n} a_k \cdot d^k. \tag{5.1.6}$$

Here we use the subscript $d$ to denote that the quantity is expressed in base $d$. For our example above we could have written $123.45_{10}$, but when no base is specified we assume base 10. Note that in order for this to make sense we must have $0 \leq a_i < d$.

An equally valid choice is base 2, known as **binary**. Then we have $0 \leq a_i < 2$, so the only options are the digits 0 and 1. Suppose we have the binary number $101.1_2$. Using our general formula we have

$$101.1_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 4 + 0 + 1 + \frac{1}{2} = 5.5_{10}. \tag{5.1.7}$$

If you were to count in binary, starting at 0, the first few numbers, 0 to 16, are

$$0_2, 1_2, 10_2, 11_2, 100_2, 101_2, 110_2, 111_2, 1000_2, 1001_2, 1010_2, \\ 1011_2, 1100_2, 1101_2, 1110_2, 1111_2, 10000_2. \tag{5.1.8}$$

**Problem 5.1.9** How high can you count in binary?
Can you convert the following to decimal[a]:

- $11010_2$;
- $101011_2$;
- $1010000_2$;
- $101.0101_2$;
- $1000.0001_2$.

Can you convert the following to binary[b]:

- 9;
- 27;
- 104;
- 12.5;
- 8.6.

[a] 26, 43, 80, 5.3125, 8.0625

[b] $1001_2$, $11011_2$, $1101000_2$, $1100.1_2$, $1000.10011001100110011001101_2$

**Application 5.1.10** When you really get down to it computers are pretty limited in what they can actually represent. A very crude model of a computer is a bunch of switches, which are all either on or off. Then all of the impressive things a computer can do are just turning these switches on and of very *very* quickly.

This makes binary the natural choice for a computer. You can store a 0 as an off switch, and a 1 as an on switch.

For example, C has several data types which can store integer values, such as `short`, `int`, `long int` (or just `long` for short). The implementation of C which I tried has these storing 16, 32, and 64 bits respectively. That is, a `short` stores its data using 16 1s and 0s. If we want to use all of these to store the value then we should use `unsigned short`, otherwise one of the bits is used up storing the sign. When we do this the largest value we can represent is $1111111111111111_2$ (that's 16 1s), which is $65\,535_{10}$:

$$1 \cdot 2^{15} + 1 \cdot 2^{14} + \cdots + 1 \cdot 2^{1} + 1 \cdot 2^{0} = 2^{16} - 1. \tag{5.1.11}$$

Similarly, the largest value that `unsigned int` can store is $11111111111111111111111111111111_2$ (32 1s), which is $4\,294\,967\,295_{10} = 2^{32} - 1$, and the largest value that `unsigned long` can store is $1111111111111111111111111111111111111111111111111111111111111111_2$ (64 1s), which is $18\,446\,744\,073\,709\,551\,615_{10}$. If you want to deal with values larger than this in C you need special data types.

We'll see later exactly how computers store these numbers, including signs.

**Remark 5.1.12** On most computers the time is stored not as a human readable time, but as Unix time. The computer stores the number of seconds since 00:00:00 UTC on January 1st 1970. For example, at the time of writing the time in Unix time is 1758619784. Here's how to get this in Python and convert it to a human-readable format:

```
>>> import time
>>> time.time()
1758619784
>>> time.localtime(1758619784)
time.struct_time(tm_year=2025, tm_mon=9, tm_mday=23,
    tm_hour=10, tm_min=29, tm_sec=44, tm_wday=1,
    tm_yday=266, tm_isdst=1)
```

So the time is 10:29:44 BST.

On many computers this is stored as a 32-bit signed integer. The sign allows computers to understand times before 1970. It takes one bit to store the sign. This leaves 31 bits for storing the number. The largest number that can be stored is then $2^{31} - 1 = 2\,147\,483\,647$. It turns out that 2 147 483 647 seconds after the start of Unix time is 03:14:07 UTC January 19th 2038. There is some worry that at this point any computer relying on Unix time with 32-bit signed integers will break, since apart from telling us the time the internal clocks of computers are very important for all sorts of things. Fortunately, almost all modern computers now store the time as a signed 64-bit integer, and the largest value that can be stored is $2^{63} - 1 = 18\,446\,744\,073\,709\,551\,615$, and 18 446 744 073 709 551 615 seconds after the start of Unix time is in about 292 billion years, which is about 21 times the age of the universe. Thus, while the problem of running out of storage for the time isn't really fixed it's definitely not our problem.

Binary is a useful choice for computers, but you may have noticed that it makes even fairly small numbers take up a lot of digits. Sometimes it's useful to make a choice where we use fewer digits to store a given number. One common choice for this is hexadecimal, which is base 16. We choose 16 because it's larger than 10 and is a power of 2, which means it interacts nicely with binary.

There's a problem though. We have the digits 0123456789, but for base 16 our digits are between 0 and 15, so if we want a single symbol per digit we have to make up some new symbols for 10 to 15. There's a standard choice, we use $\mathrm{A} = 10$, $\mathrm{B} = 11$, $\mathrm{C} = 12$, $\mathrm{D} = 13$, $\mathrm{E} = 14$, and $\mathrm{F} = 15$.

For example, consider the hexadecimal $7\mathrm{AF}_{16}$. We can convert this to decimal as follows:

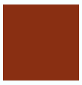
$$7\mathrm{AF}_{16} = 7 \cdot 16^2 + \mathrm{A} \cdot 16^1 + \mathrm{F} \cdot 16^0 = 3 \cdot 16^2 + 10 \cdot 16^1 + 15 \cdot 16^0 = 1967_{10}. \tag{5.1.13}$$

**Application 5.1.14** If you magnify a computer screen you'll see that a pixel is actually made out of three lights, one red, one green, and one blue. To change the colour we simply turn these lights on to different brightnesses. As discussed computers like to use binary, and a fairly common standard is that the brightness of one of these lights is measured from 0 (off) to $255 = 2^8 - 1$ (fully on). This way each pixel's colour is controlled by three 8-bit numbers, for a total of $256^3 = 16\,777\,216$ colours!

However, reading and working with binary as a human isn't great. Fortunately, $2^8 = 16^2$, so we can neatly represent our 8 bits with two hexadecimal digits. For example, if we take this red colour it's represented by #B30C00 (the # symbol is commonly used to mean the number is hexadecimal). This isn't actually one number, it's three, $\mathrm{B3}_{16}$, $\mathrm{0C}_{16}$ and $00_{16}$, representing the amount of red, green, and blue respectively. That is, we have set the red channel to $\mathrm{B3}_{16} = 11 \cdot 16^1 + 3 \cdot 16^0 = 179_{10}$ out of $255_{10}$, which is about 70 %. We've set the green channel to $\mathrm{C}_{16} = 12 \cdot 16^0 = 12_{10}$ out of $255_{10}$, which is about 4 %. Finally, we've set the blue channel to $0_{16} = 0_{10}$ out of $255_{10}$, which is to say, it's off.

Note that the brightest we can set any single colour channel is $\mathrm{FF}_{16} = 15 \cdot 16^1 + 15 \cdot 16^0 = 255$. White is then #FFFFFF, and black is #000000.

**Problem 5.1.15** Consider the colour #23F089, which of the following do you think this is[a]?

1. 

2. 🟩

3. 🟦

You don't need to do a very accurate calculation, just consider which of the three channels, red, green and blue, will be brightest.

[a] For the colour blind: the three colours are a muddy red, light green, and blue respectively.

**Code 5.1.16** Computers can do a lot of the work of converting bases for you. In Python the built-in function `int` takes a `str` as the first argument and a base to interpret the string in in the second argument[a]. The output is an `int`, which is output in base 10. The base can be any number between 0 and 36, and when we run out of digits the numbers A through Z are used for $10_{10}$ through $35_{10}$. Python also has the built-in functions `bin`, `oct` and `hex` which take an `int` and output a string representing that integer in base 2, 8, and 16 respectively. Note that Python, and a lot of other languages, denote that a number is in base 2, 8, or 16 by prefixing it with `"0b"`, `"0o"`, or `"0x"` respectively.

```
>>> int("111", 2)
7
>>> int("F3", 16)
243
>>> int("22", 8)
18
>>> bin(5)
"0b101"
>>> oct(32)
"0o40"
>>> hex(64)
"0x40"
```

Matlab has the function `dec2base`, which takes in an integer (in decimal) and a base, and outputs a string representing that integer in that base. It also has the functions `hex2dec`, `oct2dec`, `bin2dec`, and `dec2hex`, which take a string and convert it according to the name of the function. Mathematica has the function `BaseForm`, which takes in an integer and a base, and prints that integer in that base.

[a] Both `int` and `str` are in-built types in python, representing an integer and a string respectively. A string is a string of characters, which we write between quotation marks, something like `"Hello, World!"`. Through some Python magic `int` can also be called as a function, so it's playing a double role here, which is a bit confusing.

**Application 5.1.17** Another base that you're probably using all the time without realising it is base 60. This was used by the ancient Babylonians, and it's still present today in the fact that there are 60 minutes in an hour and $360 = 6 \cdot 60$ degrees in a full circle, each of which is divided into 60 (arc)minutes, which is each divided into 60 (arc)seconds.

One argument for using base 60 is that 60 has lots of divisors, which makes it much nicer to do maths with base 60. One argument against using base 60 is that you need 60 distinct digits, which is a lot of symbols to learn.

The time as I'm writing this is 10:05:43. We can interpret this as the number of seconds since midnight in base 60. It's been $10 \cdot 60^2 + 5 \cdot 60^1 + 43 \cdot 60^1 = 36343$ seconds since midnight. Here we get around the 60-digit problem by using a base 10 representation of each digit.

**Application 5.1.18** Youtube video IDs are 11 characters long, and can use any of the following characters:

$$\text{0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ}$$
$$\text{abcdefghijklmnopqrstuvwxyz+/} \tag{5.1.19}$$

There are 64 characters here, so hopefully at this point it's no surprise that actually the Youtube video ID is a number in base 64.

Speaking of Youtube, here's the video I learned this from. The question posed in the video is if we'll ever run out of video IDs? Have a go at answering this before you watch the video. [Hint: What's the largest number you can make in base 64 with 11 digits?]

Here's a general formula for writing a number, $n$, in base $d$.

1. Find the largest power of $d$ less than $n$, say $n = md^k + r$ where $0 \le r < d^k$. Your first digit is then in position $k$ (with the units being position 0), and it is whatever digit represents $m$.
2. If $r \neq 0$ then set $n = r$ and repeat.
3. If $r = 0$ stop.

**Example 5.1.20** uppose we want to write $1310_{10}$ in hexadecimal. Then $n = 1310_{10}$ and $d = 16$. The first few powers of 16 are 16, $16^2 = 256$, and $16^3 = 4096$, which is larger than 1310, so we can stop there. Then we look to write

$$1310 = m \cdot 256 + r \tag{5.1.21}$$

with $0 \le r < 256$. Rearranging we have $r = 1310 - m \cdot 256$. For $m = 1$ we get $r = 1054$, for $m = 2$ we get $r = 798$, for $m = 3$ we get $r = 542$, for $m = 4$ we get $r = 286$, and for $m = 5$ we get $r = 30$, which is in the required range. Thus, our first digit goes in the $k = 2$ position (remembering that units is position 0). This digit should be $m = 5$. Thus, our number is $5xy_{16}$, where

we still have to determine $x$ and $y$.
Since $r = 30 \neq 0$ we repeat with $d = 30$. The largest power of 16 less than this is just $16^1 = 16$. Then $d = 1 \cdot 16^1 + 14$, so our next digit is 1, and it goes in position 1 (remembering that the units is position 0). Thus, our number is $51y_{16}$, with $y$ still to be determined.
Since $r = 14 \neq 0$ we repeat with $d = 14$. The largest power of 16 less than this is just $16^0 = 1$. Then $d = 14 \cdot 16^0 + 0$, so our next digit is 14, which in base 16 is E. This goes in position 0, which is the units position. Thus, our final number is $51\mathrm{E}_{16} = 1310_{10}$.

**Remark 5.1.22** Notice that this algorithm is very similar to a couple of other algorithms you may know, such as the long division algorithm and polynomial division algorithm, or Euclid's algorithm for finding a greatest common divisor.
The reason all of these are so similar is that they're all exploiting the same maths of divisibility. This leads to the idea of a Euclidean domain, being any set of numbers where a Euclidean-like algorithm applies. These are important in many areas of mathematics, such as number theory.

## 5.2 Error Propagation

Suppose you have an exact value, say $\pi$. Whenever we use this in practice we only use it to finite precision, say 3.14. The question is how much of a difference does that extra 0.0015926 … make to our final calculation? The answer depends on exactly what calculation we're doing.

When a value is stated we should always state the error, unless it's implied. For example, if we know our measurement of $\pi$ is accurate to within 0.002 then we might write it as $3.14 \pm 0.002$, which means the true value is somewhere between 3.138 and 3.142.

The implied error on any quantity is plus or minus half a unit of the smallest significant figure. For example, if we just write 3.14 then the smallest significant figure is the 0.04, and one unit in this position is 0.01, and half of this is 0.005. Thus, our value is $3.14 \pm 0.005$. By which we mean that the true value is between $3.14 - 0.005 = 3.135$ and $3.14 + 0.05 = 3.145$.

Suppose we have a value, $a$, for which we only know the approximate value, $a_0$. This may be due to rounding, or maybe $a_0$ is the output of some experiment (which can never give us a value to complete accuracy). The **error**, $\varepsilon_a$, in $a_0$ is the difference from the true value, that is

$$\text{error} =:= a - a_0. \tag{5.2.1}$$

We can rewrite this as

$$a = a_0 + \text{error}. \tag{5.2.2}$$

The **absolute error** or **error modulus** is $\varepsilon_a = |\text{error}|$. Then

$$a = a_0 \pm \varepsilon_a. \tag{5.2.3}$$

The ± reflects the fact that we don't know if the value $a$ is an overestimate or underestimate of the true value.

If we have another value, $b$, and approximant, $b_0$, such that $b = b_0 \pm \varepsilon_b$ then what is the error in the quantity $c = a + b$?

We always think worse-case-scenario in these cases. This is important when engineering, we want to assume the worst case scenario because if we get this wrong then things can go horribly wrong. It's almost always better (but usually more expensive) to over-engineer things so that if the universe conspires against us and all of the errors combine to the largest possible error then whatever we've designed should still be safe. To save money then we want to know what is the absolute minimum we can over-engineer things so that we are always safe from the maximum error, but not more.

The worst case scenario is that both the error of $a$ and the error of $b$ have the same sign. When computing $c$ we have

$$c = a + b = a_0 + \varepsilon_a + b_0 + \varepsilon_b = (a + b) + (\varepsilon_a + \varepsilon_b) = c_0 + \varepsilon_c \tag{5.2.4}$$

where $c_0 = a_0 + b_0$ is our approximate value of $c$ and $\varepsilon_c = \varepsilon_a + \varepsilon_b$ is the error.

We can compute the error in calculating $d = a - b$ in a similar way. Here the worst case scenario is that the errors have opposite signs, and then

$$d = a - b = a_0 + \varepsilon_a - (b_0 - \varepsilon_b) = (a_0 - b_0) + (\varepsilon_a + \varepsilon_b) = d_0 + \varepsilon_d \tag{5.2.5}$$

where $d_0 = a_0 - b_0$ is our approximate value of $d$ and $\varepsilon_d = \varepsilon_a + \varepsilon_b$ is the error.

Combining these two results we get the following.

> **Proposition 5.2.6** The absolute error of the sum or difference of two values is the *sum* of the absolute errors of the two values.

The next obvious question is what is the approximate error in $p = ab$? Well, we have

$$p = ab = (a_0 \pm \varepsilon_a)(b_0 \pm \varepsilon_b) = a_0 b_0 \pm a_0 \varepsilon_b \pm b_0 \varepsilon_a \pm \varepsilon_a \varepsilon_b. \tag{5.2.7}$$

We can't use this in the same way as above because we have terms, such as $a_0\varepsilon_b$, which mix the measured value and the error.

If we assume[1] that $\varepsilon_a$ and $\varepsilon_b$ are small relative to $a_0$ and $b_0$ then $\varepsilon_a\varepsilon_b$ is *very* small compared to $a_0b_0$. Under this assumption we can drop the $\varepsilon_a\varepsilon_b$ term, giving

$$p \approx a_0 b_0 \pm a_0 \varepsilon_b \pm b_0 \varepsilon_a. \tag{5.2.8}$$

From this we see that the error in $p$ is approximately

$$\varepsilon_p \approx a_0 \varepsilon_b + b_0 \varepsilon_a, \tag{5.2.9}$$

where we've assumed the worst case scenario in which all the signs are the same. Dividing this quantity by $a_0b_0$ we get

$$\frac{\varepsilon_p}{a_0 b_0} = \frac{\varepsilon_p}{p_0} \approx \frac{\varepsilon_b}{b_0} + \frac{\varepsilon_a}{a_0}. \tag{5.2.10}$$

[1] If this isn't a valid assumption then our errors are too large, if an error is similar in value to the measured result then the measured result is incredibly inaccurate.

We define the **relative error** in $a$ to be

$$r_a = \frac{\varepsilon_a}{a_0}, \tag{5.2.11}$$

and similar for $b$ and $p$. Then we have

$$r_p = r_a + r_b. \tag{5.2.12}$$

A similar calculation shows that the relative error in $q = a/b$ is

$$r_q = r_a + r_b. \tag{5.2.13}$$

**Proposition 5.2.14** The relative error of the product or difference of two values is the *sum* of the relative errors of the two values.

Note that relative error is defined as a ratio of $\varepsilon_a$ and $a_0$, both of which have the same units. Therefore the relative error is dimensionless.

The **percentage error** in $a$ is

$$100 \cdot r_a = 100 \cdot \frac{\varepsilon_a}{a_0}. \tag{5.2.15}$$

The percentage error also adds when we multiply or divide two quantities, it's just that percentage errors are larger numbers, and therefore usually nicer to work with.

**Example 5.2.16** Consider a simple electric circuit consisting of a 150 Ω resistor in parallel with a volt meter. The current through such a circuit is given by Ohm's law, $V = IR$.

The resistor has a silver band, marking it as accurate to within 10 %. The volt metre measures to the nearest 0.1 V.

When a voltage of 1.5 V is measured what is the smallest value the voltage could actually be?

The current is computed by $I = V/R$, so we need to add the relative or percentage errors.

The relative error of the resistance is 0.1, which is just 10 % as a decimal.

Since the voltmeter doesn't come with a specified error it's assumed that it is ±0.05 V. The relative error in the measured voltage is therefore

$$\frac{0.05\,\mathrm{V}}{1.5\,\mathrm{V}} = 0.0333\ldots. \tag{5.2.17}$$

Thus, the total relative error is

$$r_I = r_V + r_R = 0.0333\cdots + 0.1 = 0.1333\ldots. \tag{5.2.18}$$

The computed current is

$$I_0 = \frac{V_0}{R_0} = \frac{1.5\,\mathrm{V}}{150\,\Omega} = 0.01\,\mathrm{A} \tag{5.2.19}$$

The absolute error is then

$$\varepsilon_I = r_I \cdot I_0 = 0.1333\cdots \cdot 0.01\,\mathrm{A} = 0.001333\ldots. \tag{5.2.20}$$

Thus, the final value of the current should be stated as

$$I = (0.01 \pm 0.001)\mathrm{A}. \tag{5.2.21}$$

Notice that we've rounded the error.
The smallest that the current could be expected to actually be is then $(0.01 - 0.001)\mathrm{A} = 0.009\,\mathrm{A}$.

## 5.3 Computer Arithmetic

Choose your favourite piece of software which implements a `float` type (such as Python). Type in 0.1 + 0.1 + 0.1, and evaluate. There's a good chance that the answer you get isn't the expected 0.3, but instead 0.30000000000000004. The reason for this is something known as **floating point error**. This is a result of how computers store numbers.

An arbitrary real number takes up an infinite amount of space. We have to store every single decimal digit. This is, of course, impractical. The solution, as is often the case in engineering, is that we all agree to follow a standard way of dealing with the problem. The standard in question is IEEE 754. The IEEE stands for the Institute of Electrical and Electronics Engineers. It sets out a method for approximating real numbers, with the goal of being able to represent a large range of values to a reasonable degree of accuracy.

Suppose that you want to store a real number, $x$. First, decide how many bits you want to use. The standard choice is 32. Next, we assign one of these bits to track the sign. This leaves us with 31 bits. Then we write the number as

$$x = (-1)^{\text{sign}} \cdot m \cdot b^{e} \tag{5.3.1}$$

where $m$ is called the mantissa or significand, $b$ is the base, usually 2, and $e$ is the exponent. Typically we limit the mantissa to 23 bits, leaving 8 bits for the exponent.

The exact numbers here may differ between implementations, the standard just fixes the form of the expression above and gives some common choices for how many bits to assign to each number, and also what bases to use.

**Example 5.3.2** Suppose we want to express $\pi$ in this standard. For simplicity we'll use $b = 10$ as the base, and we'll limit the mantissa to 4 decimal places. Then we write this as

$$(-1)^0 \cdot 0.3142 \cdot 10^1. \tag{5.3.3}$$

Here the sign bit is 0 because the value is positive, the mantissa is 0.3142 (having rounded to 4 decimal places) and the exponent is 1.

**Remark 5.3.4** The IEEE 754 standard also states the existence of a few things which aren't real numbers. In particular, it's possible to represent $\pm\infty$, `NaN` (which stands for not a number, and usually signifies an error or missing data), and $-0$, which is distinct from 0.

```
>>> 0.0 is -0.0
False
>>> 0.0 == -0.0
True
```

To represent these set all exponent bits to 0. Then if all mantissa bits are set to 0 it's $(-1)^{\text{sign}}\infty$, and otherwise it's `NaN`.

This format is great, it lets us represent a lot of numbers. For example, if we take the choices above then the largest value we can represent is very large, approximately $3 \times 10^{38}$.

This complicated format is chosen because it gives us more precision for smaller values, where small differences matter, and less precision for larger values, where the percentage error of the same absolute error is much smaller.

However, this format isn't perfect. Some numbers cannot be represented exactly, and these errors can propagate. This is exactly what happens with the example of

$$0.1 + 0.1 + 0.1 = 0.30000000000000004. \tag{5.3.5}$$

There's a small error in the approximate value of 0.1 which is stored by the computer, which is why the computer can output $0.1 + 0.1 = 0.2$ correctly. However, when we add the third 0.1 the combined error is large enough that it isn't rounded away.

If you're just using the computer as a calculator this is fine. Just be aware that errors like this arise and ignore the clearly incorrect extra $4 \times 10^{-16}$. This is more of a problem in, for example, a simulation, in which you might repeat the same calculations millions of times over. Then these small errors can compound and in the worst case scenarios they completely swamp the real output and your simulation produces nonsense.

# Six

# Functions

## 6.1 Function Machines

You may be familiar with the idea of a “function machine”. This is a picture for working with functions, which we’ll define shortly. The idea is that a function is a piece of machinery which takes in an input and puts out an output. We draw the function machine as follows, where $f$ is the name of the function[1].

$$\text{Input} \longrightarrow \boxed{f} \longrightarrow \text{Output} \tag{6.1.1}$$

We have to specify what inputs and outputs are allowed. Most commonly we’ll take our sets of allowed inputs and outputs to be some subsets of $\mathbb{R}$, but they can be any sets.

For example, if our allowed inputs and outputs are integers it might be that when we give $f$ the input 3 it always outputs 7. Then we could write

$$3 \longrightarrow \boxed{f} \longrightarrow 7 \tag{6.1.2}$$

There are two important things to keep in mind:

1. For a given input the output of a function is always the same, there’s no way the input 3 will be sent to the output 7 and then later to the output 42[2].

2. A function is determined by where it sends *all* allowed inputs. For example, consider two functions which have integers as possible inputs and outputs. Say both functions send 3 to 7, this *does not* mean they are the same function. For example, the first could be the function which sends any input to 4 plus that input, and the second could be the constant function which sends any input to 7. These are not the same. It is not enough to consider the function at one value, we have to consider it at all inputs.

> **Remark 6.1.3** The notation of “function machines” can actually be made rigorous using graphical calculus. This also allows this notation to be far more general than just describing functions between sets.

[1] We typically call functions $f$, then $g$, $h$, and so on. When a function represents a particular quantity it’s more common to give it a name that reflects this. For example, the function which takes in a time and outputs the acceleration of a ball at that time might be called $a$ for acceleration.

[2] It is possible to have time dependent functions, but then the time is part of the input, so the input changes if we consider it at some later time.

## 6.2 Definition

A function is a mapping of inputs to outputs. If the function is called $f$ and the input is $x$ then we call the corresponding output $f(x)$. The allowed inputs and possible outputs are part of the data of the function. If we change the allowed inputs then it changes the function.

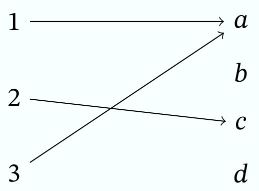
**Definition 6.2.1 — Function** Let $A$ and $B$ be sets. A **function**, $f: A \to B$ (or $A \xrightarrow{f} B$) sends elements of $A$, the **domain**, to elements of $B$, the **codomain**.

Sometimes the word "range" is used for the codomain. This should be avoided as the word range is *also* used for a specific subset of the codomain.

**Example 6.2.2** We can define a function between finite sets by specifying the output for all possible inputs. For example, if we have a function

$$f: \{1, 2, 3\} \to \{a, b, c, d\} \tag{6.2.3}$$

we can specify that $f(1) = a$, $f(2) = c$, and $f(3) = a$.
We can draw this as follows:



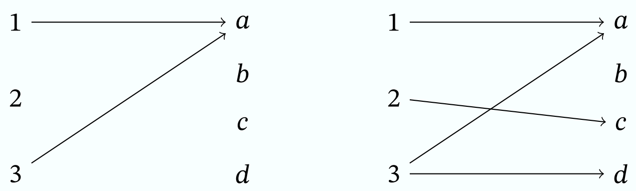
(6.2.4)

Notice that every input has exactly one arrow coming out of it. It must have at least one arrow, otherwise the function isn't defined at that point[a], and it can only have one because a function can't take on multiple values for the same input[b].
However, it is fine that there are outputs with no arrows going to them, and outputs with multiple arrows going to them. Later we'll introduce the concepts of surjective and injective functions which don't allow this.

[a] Sometimes people talk of partially defined functions. These are *not* functions according to this definition.

[b] Sometimes people talk of multi-valued functions, particularly with complex numbers. These are *not* functions according to this definition.

**Example 6.2.5 — Non-Functions** The following pictures *do not* define functions from $\{1, 2, 3\}$ to $\{a, b, c, d\}$. Can you see why?



(6.2.6)

In the first the output for 2 isn't defined, and for the second 3 is mapped to two different outputs. Neither of these things is allowed.

**Example 6.2.7** Consider the set $\{y\}$. There is exactly one function $f : A \to \{y\}$ for any set $A$. We always have to define $f(x) = y$ for any $x \in A$, there's no other choice[a].

[a] This property is called being initial.

**Example 6.2.8** We can define a function

$$T : \{\text{positions in the room}\} \to \mathbb{R} \tag{6.2.9}$$

by defining $T(p)$ (for $p$ a position) to be the temperature in the room at position $p$. This defines $T(p)$ for all possible inputs, and it does so uniquely (although perhaps in a way that changes over time), so it defines a function.

Physicists would call a function of position a field. Another example is the gravitational field, which measures how strong gravity is at a given position.

**Remark 6.2.10** Many programming languages have a notion of a function. These are often similar to the notion of a function in mathematics.

For example, in Python we might define a function which takes in a list and returns its length:

```python
def length(l: list) -> int:
   # I'm cheating, len is a built in function
   # which returns the length
    return len(l)
```

This defines a function, `length`, from the set of all lists to $\mathbb{Z}$ (although of course it's not possible to get a negative output, but the `int` type does allow for it). In Python you can give the allowed inputs and outputs with type hints.

However, these functions can often have side-effects, such as changing the value of some global variable.

```python
x = 3
def length(l: list) ->:
    # the global keyword lets us change
    # the value of x out of the scope
    global x
    x = 4
    return len(l)
```

Running this function changes the value of `x` from 3 to 4. For this reason we call them impure functions. This is *not* a function as mathematicians would define it. Some languages, like Haskell, don't allow this, and so in Haskell functions are functions as mathematicians would define them. In Haskell the notation for defining a function is very similar to the notation for defining a function in maths. For example, the factorial function may be defined as

```haskell
fac :: Int -> Int
fac 0 = 1
fac n = n * fac (n - 1)
```

The main difference is that Haskell doesn't use as many brackets, so we write `fac x` instead of `fac(x)`.

Often the inputs and outputs of our function are some type of number. Then we can often specify the output of our function via a formula. For example, we might define a function, $f \colon \mathbb{R} \to \mathbb{R}$, by declaring that $f(x) = 3x + 1$. That is, whatever the input of our function is the output is the result of multiplying the input by 3 and then adding 1. Another example might be to define a function

$$g \colon \mathbb{Z} \setminus \{0\} \to \mathbb{Q} \tag{6.2.11}$$

$$n \mapsto \frac{1}{n}. \tag{6.2.12}$$

Here we use the arrow "$\mapsto$" with a bar at one end to mean that if the input is $n$ the output is $1/n$. This is exactly the same as writing $g(n) = 1/n$, but sometimes the arrow notation is clearer.

I cannot stress enough that a formula defining a function and the function itself are *not the same thing*. First, not all functions are defined by a formula. Second, the formula doesn't carry as much information as the function does. In particular, the formula doesn't know about the domain and codomain of the function (with the possible exception that from the formula you can often exclude some values from the inputs, such as how we exclude 0 above). For example, we can make the definitions

$$f \colon \mathbb{R} \to \mathbb{R} \qquad\qquad g \colon \mathbb{Z} \to \mathbb{Z} \tag{6.2.13}$$

$$x \mapsto 3x + 1, \qquad\qquad x \mapsto 3x + 1. \tag{6.2.14}$$

These are defined by the same formula, but they have different (co)domains, and so are different functions. It doesn't make sense to try to compute, say, $g(1/2)$, since $1/2 \notin \mathbb{Z}$, even though $3(1/2) + 1$ makes perfect sense. On the other hand we can compute $f(1/2)$, and we find that it is $3(1/2) + 1 = 5/2$.

When you define a function by a formula you should always check two thins:

1. That allowed inputs of the function make sense in the formula (check for things like division by zero or square rooting a negative);

2. That the possible outputs of the formula given the allowed inputs of the formula are in the codomain.

Here are examples failing each of these. First, if we try to define $f : \mathbb{R} \to \mathbb{R}$ by $f(x) = 1/x$ then this isn't valid, since 0 is an allowed input and $f(0) = 1/0$ is not defined, and certainly isn't a real number. Second, if we try to define $g : \mathbb{Z} \to \mathbb{Z}$ by $g(n) = \sqrt{n}$ then this isn't allowed because, for example, 2 is a valid input but $\sqrt{2} \notin \mathbb{Z}$.

## 6.3 Injective Functions

Consider the function $f : \mathbb{R} \to \mathbb{R}$ given by $f(x) = x^2$. We have $f(-2) = (-2)^2 = 4$ and $f(2) = 2^2 = 4$. So multiple inputs map to the same output. Sometimes we want to avoid this, and we give functions where this doesn't happen a special name.

**Definition 6.3.1 — Injective** A function, $f : A \to B$, is **injective** or **one-to-one** if two different inputs have different outputs.
In symbols, if $a, a' \in A$ are such that $a \neq a'$ then $f(a) \neq f(a')$.

The terminology "injective" is preferred by mathematicians, but the term "one-to-one" is still common, particularly informally. Do not confuse "one-to-one" with "one-to-one correspondence", which we'll define later to mean injective and surjective (also defined later). I will use injective, not one-to-one, but one-to-one does have the advantage of being more descriptive, one input goes to one output.

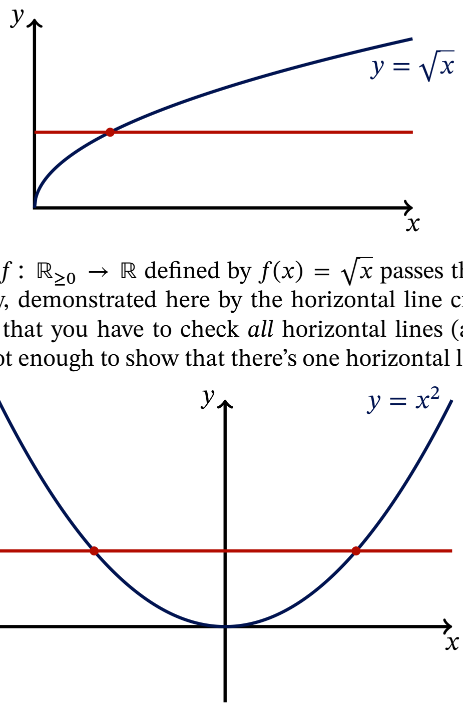
You may also hear people using two-to-one or many-to-one to describe functions which *aren't* injective. For example, $f : \mathbb{R} \setminus \{0\} \to \mathbb{R}$ is two-to-one because two inputs, such as $\pm 2$, map to each output, here 4.

**Example 6.3.2** The function $f : \{1, 2, 3\} \to \{a, b, c, d\}$ defined as follows is injective:

$$\begin{tikzcd}
1 \arrow[rr] & & a \\
 & & b \\
2 \arrow[rr] & & c \\
3 \arrow[rr] & & d
\end{tikzcd} \tag{6.3.3}$$

No two inputs go to the same output. That is, there's no output with two arrows going into it. Compare this to Equation (6.2.6), which defines a function which *isn't* injective.

For functions between intervals if we can plot them there's a nice test for injectivity, called the **horizontal line test** for injectivity. This states that the function is injective if any horizontal line we draw on the plot crosses the graph at most once.



(a) The function $f\colon \mathbb{R}_{\geq 0} \to \mathbb{R}$ defined by $f(x) = \sqrt{x}$ passes the horizontal line test for injectivity, demonstrated here by the horizontal line crossing the graph only once. Note that you have to check *all* horizontal lines (at a height in the codomain), it's not enough to show that there's one horizontal line which passes.

(b) The function $g\colon \mathbb{R} \to \mathbb{R}$ defined by $g(x) = x^2$ fails the horizontal line test for injectivity, demonstrated here by the horizontal line crossing the graph twice. Note that a horizontal line at height 0 (so along the $x$-axis) would pass the horizontal line test, crossing the graph only at $(0, 0)$. However, we need *all* horizontal lines (at a height in the codomain) to pass the test for the function to be injective.

Figure 6.1: Horizontal line test for injectivity.

! This is the horizontal line test for *injectivity*. Do not confuse it with the horizontal line test for *surjectivity* or *bijectivity*.

For example, the function $f\colon \mathbb{R}_{\geq 0} \to \mathbb{R}$ given by $f(x) = \sqrt{x}$ is injective, since it passes the horizontal line test (Figure 6.1a). The function $g\colon \mathbb{R} \to \mathbb{R}$ given by $g(x) = x^2$ is not injective, since it fails the horizontal line test (Figure 6.1b).

Note that it's not enough to just look at the function rule to determine if a function is injective. We also need to look at the domain and codomain. For example, the function $h\colon \mathbb{R}_{\geq 0} \to \mathbb{R}$ given by $h(x) = x^2$ *is* injective, because we've restricted to only consider positive inputs, which means that the $h(-2) = h(2)$ problem we had earlier doesn't arise, because it doesn't even make sense to consider $h(-2)$. This function passes the horizontal line test because in Figure 6.1b we've just chopped off the left-hand-side of the graph, which means we no longer have problems with the horizontal line crossing the graph twice.

Often it's possible to take a function which is not injective, and restrict the domain to get an injective function. For the example above we've restricted the domain from $\mathbb{R}$ to $\mathbb{R}_{\geq 0}$. This allows us to use the properties of an injective function, but we do lose some information in this restriction, here what happens to the negative numbers. There's also some choice to be made. It would be just as valid to restrict to $\mathbb{R}_{\leq 0}$ instead, but typically we want to work with positive numbers where possible, which is why I chose $\mathbb{R}_{\geq 0}$.

**Problem 6.3.4** Which of the following functions are injective[a]? If one of the functions isn't injective give an example of two elements which map to the same value.

1. $f\colon \{1, 2, 3\} \to \{a, b, c\}$ defined by $f(1) = a$, $f(2) = b$, $f(3) = a$;
2. $g\colon \{1, 2, 3\} \to \{a, b, c\}$ defined by $g(1) = a$, $g(2) = c$, $g(3) = b$;
3. $h\colon \mathbb{Z} \to \mathbb{Z}$ defined by $n \mapsto 2^n$;
4. $k\colon \mathbb{R} \to \mathbb{R}$ defined by $k(x) = 3x + 1$;
5. $y\colon \mathbb{R} \to \mathbb{R}$ defined by $y(x) = 3x^3 + 2x^2$.

[a] Ans: $g$, $h$, and $k$ are injective.

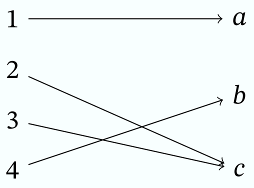
## 6.4 Surjective Functions

Consider the function $f\colon \mathbb{R} \to \mathbb{R}$ given by $f(x) = x^2$. For any positive output, $y \in \mathbb{R}_{\geq 0}$, we can get this value by $f(\sqrt{y}) = (\sqrt{y})^2 = y$. However, there's no way to get a negative output[3]. Some functions don't hit every possible output. Sometimes we want to avoid this, and we give functions where this doesn't happen a special name.

**Definition 6.4.1 — Surjective** A function, $f\colon A \to B$, is **surjective** or **onto** if all possible outputs are achieved by the function.
In symbols, if $b \in B$ then there exists some $a \in A$ with $f(a) = b$.

The terminology "surjective" is preferred by mathematicians, but the term "onto" is still common, particularly as a shorthand, people will say "$f$ is a function from $A$ *onto* $B$" to mean that $f\colon A \to B$ is a function and $f$ is surjective (note if $f$ wasn't surjective we'd say "$f$ is a function from $A$ *to* $B$").

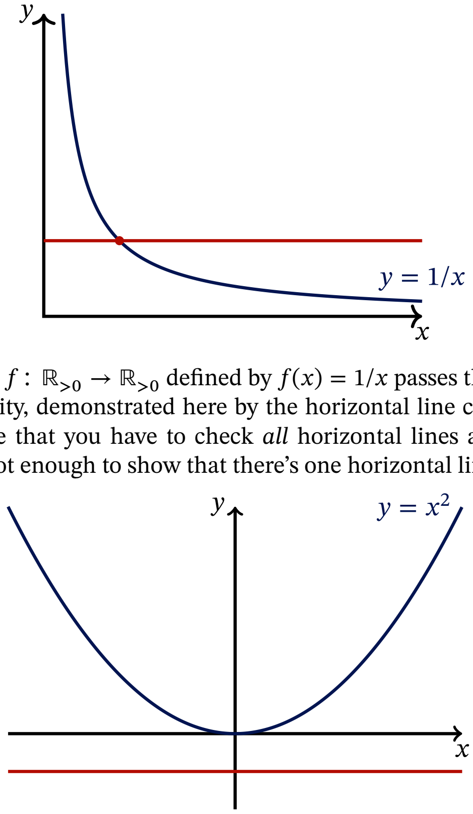
**Example 6.4.2** The function $f\colon \{1, 2, 3, 4\} \to \{a, b, c\}$ defined as follows is surjective



(6.4.3)

There is no output that we don't hit. That is, there is no output without an arrow going into it. Compare this to Equation (6.2.6), which defines a function which *isn't* surjective.

[3] In Block 2 you'll see that you can get negative numbers by squaring, if the thing you start with is imaginary. However, our function explicitly only allows *real* inputs, so that doesn't help us here.



(a) The function $f : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ defined by $f(x) = 1/x$ passes the horizontal line test for surjectivity, demonstrated here by the horizontal line crossing the graph only once. Note that you have to check *all* horizontal lines at a height in the codomain, it's not enough to show that there's one horizontal line which passes.

(b) The function $g : \mathbb{R} \to \mathbb{R}$ defined by $g(x) = x^2$ fails the horizontal line test for surjectivity, demonstrated here by the horizontal line which doesn't cross the graph. Note that some horizontal lines do pass the horizontal line test, but, we need *all* horizontal lines at a height in the codomain to pass the test for the function to be surjective.

Figure 6.2: Horizontal line test for surjectivity.

For functions between intervals if we can plot them there's a nice test for surjectivity, called the **horizontal line test** for surjectivity. This states that the function is surjective if any horizontal line we draw, at a height in the codomain, crosses the graph at most once.

! This is the horizontal line test for surjectivity. Do not confuse it with the horizontal line test for injectivity or bijectivity.

For example, the function $f : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ given by $f(x) = 1/x$ is surjective. If we have the output $y \in (0, \infty)$ then we can hit this with $f(1/y) = 1/(1/y) = y$ and if $y \in (0, \infty)$ then $y \in (0, \infty)$, so this is really in the domain. Figure 6.2a shows that this passes the horizontal line test for surjectivity. On the other hand, $g : \mathbb{R} \to \mathbb{R}$ isn't surjective, as can be seen by it failing the horizontal line test (Figure 6.2b).

Note that it's not enough to just look at the function rule to determine if a function is surjective. We also need to look at the domain and codomain. For example, the function $h : \mathbb{R} \to \mathbb{R}_{\geq 0}$ given by $h(x) = x^2$ *is* surjective, because we've restricted to only consider positive outputs, so the horizontal line at a negative value in Figure 6.2b is not a line we need to consider for the horizontal line test. Any positive output, $y \in \mathbb{R}_{\geq 0}$, can be hit by $h(\sqrt{y}) = y$.

**Problem 6.4.4** Which of the following functions are surjective[a]? If one of the functions isn't surjective give an example of an element of the codomain which the function never achieves.

1. $f: \{1, 2, 3\} \to \{a, b, c\}$ defined by $f(1) = a$, $f(2) = b$, $f(3) = a$;
2. $g: \{1, 2, 3\} \to \{a, b, c\}$ defined by $g(1) = a$, $g(2) = c$, $g(3) = b$;
3. $h: \mathbb{R} \to [-1, 1]$ defined by $h(x) = \sin x$;
4. $k: \mathbb{R} \to \mathbb{R}$ defined by $k(x) = 3x + 1$;
5. $y: \mathbb{R} \to \mathbb{R}$ defined by $y(x) = 3x^4 - 2x^2$.

[a] Ans: $g$, $h$, and $k$ are surjective.

Often it's possible to take a function which is not surjective, and restrict the codomain to get a surjective function. For the example above we've restricted the codomain from $\mathbb{R}$ to $\mathbb{R}_{\geq 0}$. This allows us to use the properties of a surjective function. Usually, when we do this we want to restrict to the largest possible subset of the codomain where everything is hit. We give this set a name.

**Definition 6.4.5 — Image** The **image** of a function, $f: A \to B$, is the set, $\operatorname{im} f$, of all values that $f$ hits. In symbols,

$$\begin{aligned} \operatorname{im} f &= \{y \in B \mid \text{there exists some } x \in A \text{ with } f(x) = y\} && (6.4.6) \\ &= \{f(x) \mid x \in A\}. && (6.4.7) \end{aligned}$$

! Do not confuse $\operatorname{im} f$ (the image of a function) with $\operatorname{Im} z$ (the imaginary part of a complex number) which you'll see in Block 2.

Some authors write $f(A)$ instead of $\operatorname{im} f$. This is slightly an abuse of notation, since $A$ is not a valid input for $f$, but it's supposed to be shorthand for $\{f(a) \mid a \in A\}$.

The image is the largest subset of $B$ where everything is hit by $f$. Thus, if we have a non-surjective function, $f: A \to B$, we can define a new function, $\tilde{f}: A \to \operatorname{im} f$, by $\tilde{f}(a) = f(a)$ which will be surjective. This is a *different* function to the original one, since the codomain is different, but it is so similar that it's common to still call it $f$, rather than $\tilde{f}$[4].

If $f: A \to B$ is surjective then every output is hit, and so $\operatorname{im} f = B$. In fact, we can take $\operatorname{im} f = B$ as the definition of surjectivity.

**Theorem 6.4.8.** A function is surjective if and only if its image is equal to its codomain.

[4] Note that $\tilde{f}$ isn't standard notation, I just made this notation up here.

## 6.5 Bijective Functions

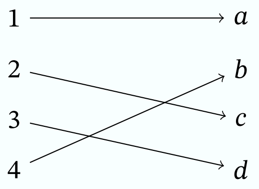
Injective functions and surjective functions are both very special cases of functions. Do you know what's even more special? Functions which are injective *and* surjective. We give these a special name too.

**Definition 6.5.1 — Bijective** A function, $f: A \to B$, is **bijective** or a **one-to-one correspondence** if it is injective and surjective.

The terminology "bijective" is preferred by mathematicians, but the term "one-to-one correspondence" is sometimes used when we want to interpret a bijective function as a pairing of elements in the following sense.

**Notation 6.5.2** We say "$f: A \to B$ is a bijection" as shorthand for "$f: A \to B$ is a function and is bijective". We use injection and surjection similarly.

**Example 6.5.3** The function $f: \{1, 2, 3, 4\} \to \{a, b, c, d\}$ defined as follows is surjective



(6.5.4)

It is injective as we don't hit any output more than once, and it is surjective because every output is hit at least once. That is, it is bijective because every output is hit exactly once. There is exactly one arrow going to every output.

Generally, if we have two sets, $A$ and $B$, and a bijection, $f: A \to B$, we get a "pairing", or one-to-one correspondence, between these sets where we pair up $a \in A$ with $f(a) \in B$. The definition of a function is exactly such that for each $a$ there is a unique $b \in B$ with $f(a) = b$ and the definition of a bijection turns this uniqueness around, for each $b \in B$ there is a unique $a \in A$ such that $f(a) = b$.

Notice that the definition of a bijection adds some symmetry to the definition of a function. With a function we can go from $A$ to $B$. A bijection allows us to go back from $B$ to $A$.

**Remark 6.5.5** A bijection between two finite sets exists if and only if the two sets have the same number of elements. This is required if we're going to construct a pairing between the sets.

Similarly, an injection, $A \to B$, exists if and only if $|A| \leq |B|$. The intuition is that there has to be enough "stuff" in $B$ for our map to hit a different thing for each input. Conversely, a surjection, $A \to B$, exists if and only if

$|A| \geq |B|$. The intuition is that there has to be enough "stuff" in $A$ to cover all of $B$. Then of course if we want a bijection, $A \to B$, we need $|A| \leq |B|$ for an injection, and $|A| \geq |B|$ for a surjection, so $|A| = |B|$.
We actually use the existence of a bijection between $A$ and $B$ to define what it means for sets to have the same size (cardinality) in the infinite case. Two sets have the same cardinality if and only if there exists a bijection between them. This is a sensible definition, but nevertheless leads to some facts which can be counter intuitive, infinity is weird:

- $|\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}|$ (we call these countably infinite because the natural numbers are the counting numbers);
- $|\mathbb{R}| > |\mathbb{Q}|$ (we say the real numbers are uncountably infinite);
- $|[0, 1]| = |\mathbb{R}|$ "there are as many numbers between 0 and 1 as there are numbers (including those numbers between 0 and 1!)".

Fun fact: There may or may not be a set larger than $\mathbb{Z}$ but smaller than $\mathbb{R}$, whether there is or not is "independent of ZFC" (this is known as the continuum hypothesis). ZFC is the standard theory of how sets work (it's complicated, we've ignored many details). Independence means that whether we assume such an in-between set exists or not maths works just as well, so we can't say either way!

Just as for injective and surjective functions if we can plot our function there is a **horizontal line test** for bijectivity. We just combine the horizontal line tests for injectivity and surjectivity. A function is bijective if any horizontal line we draw, at a height in the codomain, crosses the graph exactly once.

! This is the horizontal line test for bijectivity. Do not confuse it with the horizontal line test for injectivity or surjectivity.

We'll come back to bijections later, particularly to the idea of them providing a pairing between the domain and codomain.

## 6.6 Composition

Suppose we have functions $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$, say $f(x) = x^2$ and $g(x) = 3x+1$. Then it makes sense to apply one of these functions and then the other. For example, if we start with 3 we can apply $f$ to get $f(3) = 3^2 = 9$, then we can apply $g$ to get $g(9) = 3 \cdot 9 + 1 = 29$. More compactly, we can write $g(f(3)) = 29$. We could instead start with $g$, $g(3) = 3 \cdot 3 + 1 = 10$, then apply $f$, $f(10) = 10^2 = 100$. More compactly, we can write $f(g(3)) = 100$.

This chaining together of functions is an important operation. However, it doesn't always make sense. We need the output of the first function we apply to be a valid input of the second function. This leads to the following definition.

**Definition 6.6.1 — Composition** Let $f : A \to B$ and $g : B \to C$ be functions. Their **composite** is the function

$$g \circ f : A \to C \tag{6.6.2}$$

defined by

$$(g \circ f)(x) = g(f(x)). \tag{6.6.3}$$

It is very important that the codomain of $f$ is the domain of $g$, otherwise the composite doesn't make sense. I like to think of composition in terms of the following picture:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[rr, bend right, "g\circ f"'] & B \arrow[r, "g"] & C \end{tikzcd} \tag{6.6.4}$$

The idea is that we're defining a new function from $A$ to $C$ by going through $B$. If we don't have a common set, $B$, in the middle we can't make this definition.

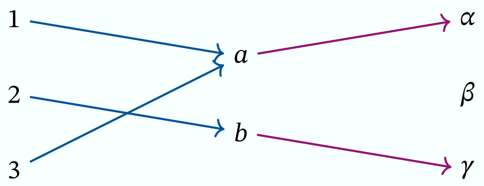
Notice that the order is important, if $A \neq C$ then we can't define $f \circ g$, even though $g \circ f$ is defined. Even if $A = C$ we saw at the start of the section that $f \circ g$ and $g \circ f$ aren't necessarily the same.

There are several ways to read "$g \circ f$". One is "$g$ composed (with) $f$". Another is "$g$ after $f$", which reminds us of the order, $g \circ f$ is the result of applying $g$ after we apply $f$. Note that we read the order backwards, applying the rightmost function first. This makes sense when we evaluate because in $(g \circ f)(x)$ the $f$ is closest to the input, so we apply that first. Another way that people may read this is "g circ $f$", the reason being that in LaTeX[5] (the typesetting software used to typeset these notes, and most of maths) the symbol $\circ$ is given by the command `\circ`.

**Example 6.6.5** Consider the functions $f : \{1, 2, 3\} \to \{a, b\}$ and $g : \{a, b\} \to \{\alpha, \beta, \gamma\}$ defined by $f(1) = f(3) = a$, $f(2) = b$, $g(a) = \alpha$ and $g(b) = \gamma$. We can form the composite $g \circ f$. We can just compute what the output should be for each input:

- $(g \circ f)(1) = g(f(1)) = g(a) = \alpha$;
- $(g \circ f)(2) = g(f(2)) = g(b) = \gamma$;
- $(g \circ f)(3) = g(f(3)) = g(a) = \alpha$.

We can draw the original functions as follows:

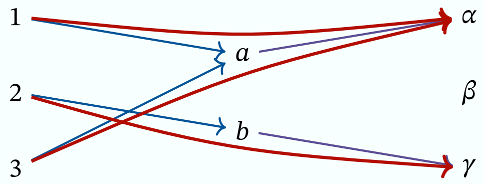


(6.6.6)

Then the composite is the function which follows each of these arrows

[5] Pronounced lay-tek

from the first set to the last skipping past the middle:



(6.6.7)

The fact that order is important means that ∘ is not commutative. It is, however, associative.

**Lemma 6.6.8** Composition of functions is associative.

*Proof.* Let $f : A \to B$, $g : B \to C$, and $h : C \to D$ be functions. Then $g \circ f : A \to C$ is a function, and we can define $h \circ (g \circ f) : A \to D$. Similarly, $h \circ g : B \to D$ is a function, and we can define $(h \circ g) \circ f : A \to D$. We now show that actually these are the same functions.
Let $x \in A$, then

$$(h \circ (g \circ f))(x) = h((g \circ f)(x)) = h(g(f(x))) \tag{6.6.9}$$

and

$$((h \circ g) \circ f)(x) = (h \circ g)(f(x)) = h(g(f(x))). \tag{6.6.10}$$

Since these are the same for all $x \in A$ we know that

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

□

## 6.7 Identity Functions

Given an arbitrary set, $A$, we really can't define any interesting functions from $A$ to itself. For example, the function rule $x \mapsto 3x + 1$ only makes sense if we have a notion of adding and multiplying in $A$, but what if $A = \{\text{apple}, \text{blue}, \diamond\}$? Then what does $3 \cdot \text{apple} + 1$ mean? Nothing.

In fact, there's only one function we can define without knowing anything, the "do nothing" function, $x \mapsto x$.

**Definition 6.7.1 — Identity Function** The **identity function** on a set, $A$, is the function

$$\begin{aligned} \mathrm{id}_A : A &\to A \\ x &\mapsto x \end{aligned} \tag{6.7.2}$$

That is, $\mathrm{id}_A(x) = x$.

This function is also sometimes called $1_A$. When $A$ is clear from context it's common to just write id. Note that if $A$ and $B$ are different sets then $\mathrm{id}_A$ and $\mathrm{id}_B$ are different functions, having different domains and codomains, even though they have the same function rule.

While the identity function is very boring on its own it's interaction with other functions, via composition, is what makes it important. Let $f : A \to B$ be an arbitrary function. Then we can form the composite $f \circ \mathrm{id}_A : A \to B$. We can work out what this function does to $x \in A$:

$$(f \circ \mathrm{id}_A)(x) = f(\mathrm{id}_A(x)) = f(x). \tag{6.7.3}$$

Since this is true for all $x \in A$ we have that

$$f \circ \mathrm{id}_A = f. \tag{6.7.4}$$

Similarly, we can form the composite $\mathrm{id}_B \circ f : A \to B$, and for $y \in B$ this acts as follows:

$$(\mathrm{id}_B \circ f)(y) = \mathrm{id}_B(f(y)) = f(y). \tag{6.7.5}$$

Since this is true for all $y \in B$ we have that

$$\mathrm{id}_B \circ f = f. \tag{6.7.6}$$

In fact, this property characterises the identity function. By this we mean that if $f : A \to B$ and $g : B \to A$ are some functions and $h : A \to A$ is a function such that

$$f \circ h = f, \qquad \text{and} \qquad h \circ g = g \tag{6.7.7}$$

then it must be that $h = \mathrm{id}_A$.

The idea here is that the identity function plays the role of 0 or 1 when it comes to composition. We have $x + 0 = x$ and $x \cdot 1 = x$, similarly we have $f \circ \mathrm{id}_A = f$. We call elements which don't really do anything in a binary operation, like these examples, identities, and this is why $\mathrm{id}_A$ is the identity function.

## 6.8 Invertible Functions

A common question we will sometimes want to ask is if we can undo a function. For example, if $f : A \to B$ for $a \in A$ we can always find $b \in B$ with $f(a) = b$, we just apply $f$ to $a$ and choose this value to $b$. Is there a way to get from $b \in B$ and find some $a \in A$ with $f(a) = b$? Sometimes yes, sometimes no.

Let's be clear about what we want to do. We want to apply a function and then undo it, which should be the same as doing nothing. The question we're asking is is there something I can compose with my function to get back the identity? This leads to the following definition.

> **Definition 6.8.1 — Invertible Function** A function, $f : A \to B$, is called **invertible** if there exists some function $g : B \to A$ such that
>
> $$g \circ f = \mathrm{id}_A, \qquad \text{and} \qquad f \circ g = \mathrm{id}_B. \tag{6.8.2}$$

Evaluating these defining properties, $f$ is invertible if and only if for all $x \in A$ and all $y \in B$ we have

$$g(f(x)) = \mathrm{id}_A(x) = x, \qquad \text{and} \qquad f(g(y)) = \mathrm{id}_B(y) = y. \tag{6.8.3}$$

**Example 6.8.4** Consider the function $f : \mathbb{R} \to \mathbb{R}$ given by $f(x) = 3x + 1$. I claim that this is invertible, with inverse $g : \mathbb{R} \to \mathbb{R}$ given by $g(x) = (x - 1)/3$. To see this is true we just need to check that $f(g(x)) = x$ and $g(f(x)) = x$. Starting with the first one we have

$$\begin{aligned} f(g(x)) &= 3g(x) + 1 && (6.8.5)\\ &= 3\frac{x-1}{3} + 1 && (6.8.6)\\ &= x. && (6.8.7) \end{aligned}$$

I computed this starting by applying the definition of $f$, but we could have started with $g$ instead, for the computation

$$\begin{aligned} f(g(x)) &= f\left(\frac{x-1}{3}\right) && (6.8.8)\\ &= 3\frac{x-1}{3} + 1 && (6.8.9)\\ &= x. && (6.8.10) \end{aligned}$$

For the second calculation we have

$$\begin{aligned} g(f(x)) &= \frac{f(x)-1}{3} && (6.8.11)\\ &= \frac{3x+1-1}{3} && (6.8.12)\\ &= x. && (6.8.13) \end{aligned}$$

Similarly, we could have started with the definition of $f$:

$$\begin{aligned} g(f(x)) &= g(3x+1) && (6.8.14)\\ &= \frac{3x+1-1}{3} && (6.8.15)\\ &= x. && (6.8.16) \end{aligned}$$

**Lemma 6.8.17** The inverse of a function, if it exists, is unique.

*Proof.* Suppose that $f : A \to B$ is a function with inverses $g : B \to A$ and $h : B \to A$. Then in particular we know that $h \circ f = \mathrm{id}_A$ and $f \circ g = \mathrm{id}_B$. Then we have

$$g = \mathrm{id}_A \circ g = (h \circ f) \circ g = h \circ (f \circ g) = h \circ \mathrm{id}_B = h.$$ □

**Notation 6.8.18** If $f : A \to B$ is invertible its inverse is unique, and so we can give it a special name. We write $f^{-1} : B \to A$ for the inverse. So $f^{-1} \circ f = \mathrm{id}_A$ and $f \circ f^{-1} = \mathrm{id}_B$.

We use the notation $f^{-1}$ because if we think of $\circ$ as multiplication then $f^{-1}$ is like $1/f$. However, $\circ$ *is not multiplication* and so this notation is suggestive only, not literal: $f^{-1} \neq 1/f$.

Perhaps the following pictures help demonstrate what the inverse does:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "\mathrm{id}_A"'] & B \arrow[d, "f^{-1}"] \\ & A \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} B \arrow[r, "f^{-1}"] \arrow[dr, "\mathrm{id}_B"'] & A \arrow[d, "f"] \\ & B \end{tikzcd} \tag{6.8.19}$$

In terms of elements here's what these triangles mean:

$$\begin{tikzcd} x \arrow[r, mapsto, "f"] \arrow[dr, mapsto, "\mathrm{id}_A"'] & f(x) \arrow[d, mapsto, "f^{-1}"] \\ & x = f^{-1}(f(x)) \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} y \arrow[r, mapsto, "f^{-1}"] \arrow[dr, mapsto, "\mathrm{id}_B"'] & f(x) \arrow[d, mapsto, "f"] \\ & y = f(f^{-1}(x)) \end{tikzcd} \tag{6.8.20}$$

A very important result is that a function has an inverse exactly when it is bijective! The key is that we can use the pairing that the bijection gives us to form an inverse function, and vice versa the existence of the inverse let's us set up a pairing. This is made formal in the proof of the following.

**Theorem 6.8.21.** A function is invertible if and only if it is bijective.

*Proof.* First, suppose that $f: A \to B$ is a bijection. Then for each $b \in B$ there exists a unique $a \in A$ with $f(a) = b$ (by the pairing interpretation). Then we can define $f^{-1}: B \to A$ by sending this $b$ to the unique such $a$. This defines an inverse since by construction we have $f^{-1}(f(a)) = f^{-1}(b) = a$ for all $a \in A$ and $f(f^{-1}(b)) = f(a) = b$ for all $b \in B$.
Second, suppose that $f: A \to B$ is invertible. We have to show that it is bijective. To do so we show it is injective and surjective. For injectivity, suppose that $x, x' \in A$ are such that $f(x) = f(x')$, then we can apply $f^{-1}$ to either side of this equality giving $f^{-1}(f(x)) = f^{-1}(f(x'))$, which then reduces to $x = x'$. So, if $x$ and $x'$ map to the same point then they must have been the same to start with, so $f$ is injective. For surjectivity, suppose that $y \in B$. Then we can take $x = f^{-1}(y) \in A$, which has the property that $f(x) = f(f^{-1}(y)) = y$, and so $f$ is surjective. Thus, $f$ is bijective. □

For example, we know that the function $f: \mathbb{R} \to \mathbb{R}$, $f(x) = 3x + 1$, has an inverse, and so we now know that this is a bijection! Similarly, we know that the map $\{1, 2, 3\} \to \{a, b, c\}$ sending 1 to $a$, 2 to $c$ and 3 to $n$ is a bijection, so it must have an inverse, and it does, we send $a$ to 1, $b$ to 3, and $c$ to 2.

Sometimes it's easier to find an inverse, sometimes it's easier to check if something is a bijection. This theorem tells us that we only need to do one of these. Note that being bijective only tells us that an inverse exists, sometimes we still have to put in the work to figure out what it already is.

## 6.9 Computing Inverses

In this section we'll look at a method for computing the inverse of a function when we know one exists. Suppose we have a function, $f : \mathbb{R} \to \mathbb{R}$, which has an inverse. We can (sometimes) find the inverse function, $f^{-1} : \mathbb{R} \to \mathbb{R}$, by the following procedure.

1. Write $y = f(x)$.
2. Solve for $x$.
3. When you have $x =$ something that something is exactly $f^{-1}(y)$.

This method can also be used (with some care) for functions between subsets of $\mathbb{R}$. You need to pick $y$ in the codomain of $f$, and at some point you may need to consider the domain to discard some solutions.

**Example 6.9.1** Consider the function $f : \mathbb{R} \to \mathbb{R}$ given by $f(x) = 3x + 1$. We start with

$$y = f(x) = 3x + 1. \tag{6.9.2}$$

Subtracting 1 from both sides we get

$$y - 1 = 3x. \tag{6.9.3}$$

Dividing by 3 we get

$$x = \frac{y-1}{3}. \tag{6.9.4}$$

So, we have that

$$f^{-1}(y) = \frac{y-1}{3}. \tag{6.9.5}$$

Often we want to use $x$ as our variable, so we just take the above and replace $y$ with $x$:

$$f^{-1}(x) = \frac{x-1}{3}. \tag{6.9.6}$$

If we start with a function which is not invertible then we can often restrict the domain and codomain until it is. We have to be careful then about what the inverse is, because it will depend on how we restrict the (co)domain.

**Example 6.9.7** Consider the function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 1}$ given by $f(x) = x^2 + 1$. This is a bijection, which you can check with the horizontal line test, or you can just believe me. We can find the inverse. Start with

$$y = f(x) = x^2 + 1 \tag{6.9.8}$$

where $y \in \mathbb{R}_{\geq 1}$. We can subtract 1 to get

$$y - 1 = x^2. \tag{6.9.9}$$

Then we can take square roots to get

$$x = \pm\sqrt{y-1}. \tag{6.9.10}$$

However, we know that $x \in \mathbb{R}_{\geq 0}$, since $x$ is a valid input for $f$, so it must be that $x \geq 0$, and so we throw away the $-\sqrt{y-1}$ root. Then we have

$$x = \sqrt{y-1} = f^{-1}(y). \tag{6.9.11}$$

We should check that this is really an inverse:

$$f(f^{-1}(y)) = f(\sqrt{y-1}) = (\sqrt{y-1})^2 + 1 = y - 1 + 1 = y; \tag{6.9.12}$$

$$f^{-1}(f(x)) = f^{-1}(x^2+1) = \sqrt{x^2+1-1} = \sqrt{x^2} = x. \tag{6.9.13}$$

Note that $x$ is positive so we're not accidentally throwing away a negative solution.

If instead we had started with the function $g\colon \mathbb{R}_{\leq 0} \to \mathbb{R}_{\geq 1}$ defined by $g(x) = x^2 + 1$, which is also a bijection, then the process would go as follows. We start with

$$y = g(x) = x^2 + 1, \tag{6.9.14}$$

then we solve this for $x$, giving

$$x = -\sqrt{y-1}, \tag{6.9.15}$$

where we've used the fact that $x \leq 0$ to throw away $+\sqrt{y-1}$. Then $g^{-1}(y) = -\sqrt{y-1}$.

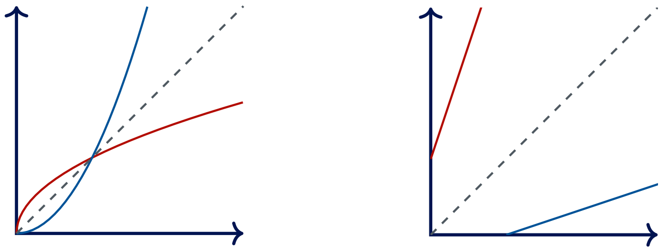
We can assume that both $f$ and $g$ arose as a result of restricting the function $h\colon \mathbb{R} \to \mathbb{R}_{\geq 1}$, $h(x) = x^2 + 1$, to a (co)domain on which it is bijective (it is already surjective, so we just need to make it injective).

Then the question is what is $h^{-1}$? The answer is it doesn't really exist, since $h$ is not bijective it is not invertible. However, we can think of $f^{-1}$ and $g^{-1}$, as being inverses of $h$ on some subset of the domain.

That is, if we want the output, $y \in \mathbb{R}_{\geq 1}$, *and* we want the input corresponding to this to be positive then we can take $x = f(y)$. If instead we still want $y$ but we want the input to be negative we can take $\tilde{x} = g(y) = -x$. This isn't a real inverse because we don't have a unique choice for the input that gives us $y$, but sometimes this is good enough. We'll see this more when we talk about inverting trigonometric functions.

Another way to compute inverses of functions from (subsets of) $\mathbb{R}$ to (subsets of) $\mathbb{R}$ is by plotting. This isn't quite as rigorous as the process above, but it can give us an idea of what the inverses is, and if we have a good guess it's often easier to check it's correct rather than go through the process of computing it from scratch.

The trick is that if we plot $y = f(x)$ it is always the mirror image of $y = f^{-1}(x)$



(a) Plotting $y = x^2$ and $y = \sqrt{x}$ we see they are mirror images in $y = x$.

(b) Plotting $y = 3x + 1$ and $y = (x - 1)/3$ we see they are mirror images in $y = x$.

Figure 6.3: Inverse functions have mirror image graphs in the line $y = x$.

in the line $y = x$. The reason for this is that plotting $y = f^{-1}(x)$ is the same as plotting $f(y) = x$ with the roles of $x$ and $y$ flipped.

# Seven

# Symmetries of Functions

In this section we'll look at two particular symmetries that a function can have. We'll start with functions being (anti)symmetric when we reflect them in the $y$-axis. Then we'll look at functions where you can shift the whole thing sideways without changing anything.

## 7.1 Even and Odd Functions