Mathematics, Reconnected: Volume 1, Foundations to Precalculus

Chris Collins
Version 0.1, 2026-10-04

Table of Contents

How to Use This Book

The Knowledge Map

Part I: Foundations

- Chapter 1. Numbers, the Number Line, and Arithmetic
- Chapter 2. Fractions, Decimals, Percentages, and Ratios
- Chapter 3. Exponents, Roots, and Scientific Notation

Part II: Algebra

- Chapter 4. The Language of Algebra: Variables, Expressions, and Equations
- Chapter 5. Linear Equations and Inequalities
- Chapter 6. Systems of Equations
- Chapter 7. Polynomials and Factoring
- Chapter 8. Quadratics

Part III: Geometry

- Chapter 9. Euclidean Geometry: Shapes, Angles, Similarity, and Proof
- Chapter 10. Coordinate Geometry

Part IV: Functions, Trigonometry, and Precalculus

- Chapter 11. Functions and Their Graphs
- Chapter 12. Exponentials and Logarithms
- Chapter 13. Trigonometry
- Chapter 14. Sequences, Series, and Summation Notation
- Chapter 15. Complex Numbers
- Appendix A: Cheat Sheet
- Appendix B: Glossary
- Appendix C: Notation and Symbols
- Appendix D: The US Mathematics Curriculum
- Appendix E: Interactive Lab Inventory

How to Use This Book

This book is meant to take you from rusty foundational mathematics to the mathematics a strong computer scientist reads and uses every day. Calculus is one milestone on that journey, not the finish line.

Who this book is for

You probably met most of this material once. Maybe it was UK sixth-form "Maths with Mechanics" decades ago, maybe an American Algebra II class. Since then, some of it has stuck and much of it has faded. When you see

$$ax^2 + bx + c = 0$$

it looks familiar, but words like *discriminant*, *vertex* or *completing the square* may not come back straight away.

So the book assumes **prior exposure with substantial gaps and rusty recall**. It won't treat you as someone who has never seen an equation. It also won't wave a section through with "you already know this". Each topic is complete enough that a real gap won't leave you stuck, and the explanations are built so you can move quickly through anything that comes back easily.

There is a second reader in mind: a teenager in the American school system. Every major topic carries a short **US curriculum** note saying roughly where an American student would meet it. If someone tells you "he's doing Algebra II", you'll know what mathematics that covers, and the explanations here can be shared with them directly.

What this book is, and is not

This book is the **canonical knowledge path**, the core of a larger project that can grow to include experiments, Python and Mathematica notebooks, interactive visualisations, and eventually a companion application with mastery tracking.

The book deliberately avoids becoming:

- **a school textbook**, padded out to fill semesters;
- **a formula encyclopedia**, where formulas are known but not understood;
- **a pure mathematics text**, where rigour comes before intuition and application;
- **a programming book in disguise**, since code reinforces the mathematics but never replaces it;
- **a calculus cram course**, which reaches derivatives fast and leaves holes underneath;
- **an endless prerequisite trap**, since you don't need to relearn everything before reaching interesting material;
- **an exercise treadmill**, because mastery matters more than volume.

How each topic is taught

Every substantial topic follows the same arc:

> *Motivation → intuition → formalisation → mechanics.*

The topic starts with a problem worth solving. Next comes what the idea **means** in plain language, then how to state it precisely, and only after that how to calculate with it efficiently. Formulas are **derived when derivation illuminates**, and theorems are **proved when proof teaches**. Procedures are shown **step by step with nothing hidden**, then summarised as algorithms, because much of mathematics really is algorithmic and you already think in algorithms.

The book keeps returning to four representations of the same idea:

words ⟷ symbols ⟷ pictures ⟷ computation

Being able to move between them is much closer to understanding than memorising any one of them.

It also keeps asking two questions: *Where would I use this?* and *When should I use this, and when shouldn't I?* Knowing when a technique fails is part of knowing the technique.

The boxes

You will see the same kinds of box throughout the book. Here is what each one is for.



Refresher boxes

Short definitions of vocabulary and notation, written to be scanned quickly. Together they form the glossary and the cheat sheet in the appendices: the "I remember learning this, remind me what it means" reference.

Example 1. Procedure boxes

Input: a technique you have just understood.

Steps: the technique written as an explicit algorithm.

Output: something you could translate into code, and often the book does.



Watch-out boxes

Failure modes, common mistakes and limitations. These are not footnotes; they are part of understanding the idea.



Interesting boxes

Where a word came from, who solved the problem first, or an unexpected modern use. Kept short.



Interactive Lab boxes

Places where a hands-on experiment would help, such as dragging a point, changing a matrix or picking a starting value. They are collected in Appendix E as the design inventory for a future companion app.

Computer-science connection sidebars

Where the mathematics shows up in computing: algorithms, graphics, databases, machine learning, information retrieval, numerical software.

US curriculum sidebars

Roughly where an American student would meet the topic. Treat these as a cross-reference, not a rule; it varies by state, district and track.

Test out where possible; slow down where necessary

There is no big entrance exam. Instead, each chapter ends with a **mastery check**: a few problems and a **decision rule**. If they are easy, move on. If they are hard, that tells you exactly which foundation needs more time, and the decision rule points you back to the relevant section.

Each chapter also opens with a **Where this chapter sits** sidebar that lists its prerequisites. If a mastery check exposes a gap, follow those links backwards.

Understand, calculate, implement, experiment

Computation in this book follows a fixed order:

1. understand the idea;
2. calculate a few cases by hand;
3. implement it (usually in a few lines of Python);
4. experiment: break it, vary it, plot it.

Asking a computer algebra system for an answer is not understanding. Writing ten lines of Python *after* doing three iterations by hand is a good way to make the understanding stick. All code uses only Python's standard library and NumPy.

The companion code

The `companion/` folder that accompanies this book has one Python folder per chapter (`ch01_numbers/` to `ch33_horizons/`), and each one follows that same order:

- `walkthrough.py` reproduces the chapter's worked examples line by line, with the reason for each step, then implements the procedure and experiments with it, including the ways it fails;
- `exercises.py` holds functions for you to implement, and `test_exercises.py` tells you when you've got them right; `solutions.py` has reference answers with commentary;
- `lab_*.py` builds the chapter's Interactive Labs as interactive plots you can drag and adjust;
- `python3 -m mathbook.mastery chNN` runs the chapter's mastery check as a quiz and tells you which sections to revisit.

A good rhythm is to read a section, run the matching part of the walkthrough, and do the exercises before moving on. See `companion/README.md` for setup.

The three volumes

The book comes in three volumes that follow the knowledge map:

Volume	Contents	Chapters
1: Foundations to Precalculus	Numbers, algebra, geometry, functions, trigonometry, sequences, complex numbers. Everything the other two volumes build on.	1–15 (Parts I–IV)
2: Continuous Mathematics	Calculus, linear algebra, numerical mathematics.	16–20, 26–28, 32 (Parts V, VII, IX)
3: Discrete and Probabilistic Mathematics	Logic, proof, number theory, combinatorics, graphs, probability, statistics, and the Horizons capstone.	21–25, 29–31, 33 (Parts VI, VIII, X)

Volumes 2 and 3 both build on Volume 1 but not on each other, so you can read them in either order or side by side. Chapter numbers are the same in every edition, which is why Volume 2 jumps from Chapter 20 to Chapter 26: a reference such as "Chapter 17" always means the same chapter, and so does the companion folder `ch17_derivatives/`. A reference to a chapter in another volume names the volume, as in "Volume 2, Chapter 17". In the PDF it links to that chapter in the online edition.

You are reading Volume 1. Each volume has the full cheat sheet, notation guide and US-curriculum map, plus a glossary and Interactive Lab inventory covering its own chapters.

How to read the rest of the book

The chapters are numbered, but the book is not a straight line. The Knowledge Map shows how the topics depend on one another. Discrete mathematics can run alongside algebra. Probability can begin early. Linear algebra does not need calculus. Choose your route, use the mastery checks, and go back when you need to.

Building the book

The source is AsciiDoc. From the repository root:

```shell
make            # regenerate all diagrams, then build build/book.html
make html       # just the HTML
make check      # fail on any AsciiDoc warning
```

Mathematics is typeset with LaTeX notation (`:stem: latexmath`) and rendered by MathJax in the HTML output. Diagrams are SVGs generated from the scripts in `diagrams/` , so they can be regenerated and changed.

The Knowledge Map

Mathematics isn't a ladder. It's a web of ideas, each depending on a few others. This chapter is that web: the route-planner for the rest of the book, and the skeleton of the future skill tree.

Why a map rather than a list

School mathematics in the US is usually taught as a sequence of year-long courses:

Pre-Algebra → Algebra I → Geometry → Algebra II → Precalculus → Calculus

That sequence is a scheduling decision, not a law of nature. The real constraint is **dependency**: you can't understand the derivative without limits, and you can't understand limits without functions. But you **can** learn graph theory long before calculus, and elementary linear algebra needs no calculus at all.

The map below shows each chapter and the chapters it builds on directly. An arrow from A to B means "B uses ideas from A". The dashed arrow marks a dependency that only matters for the optional parts of a chapter.

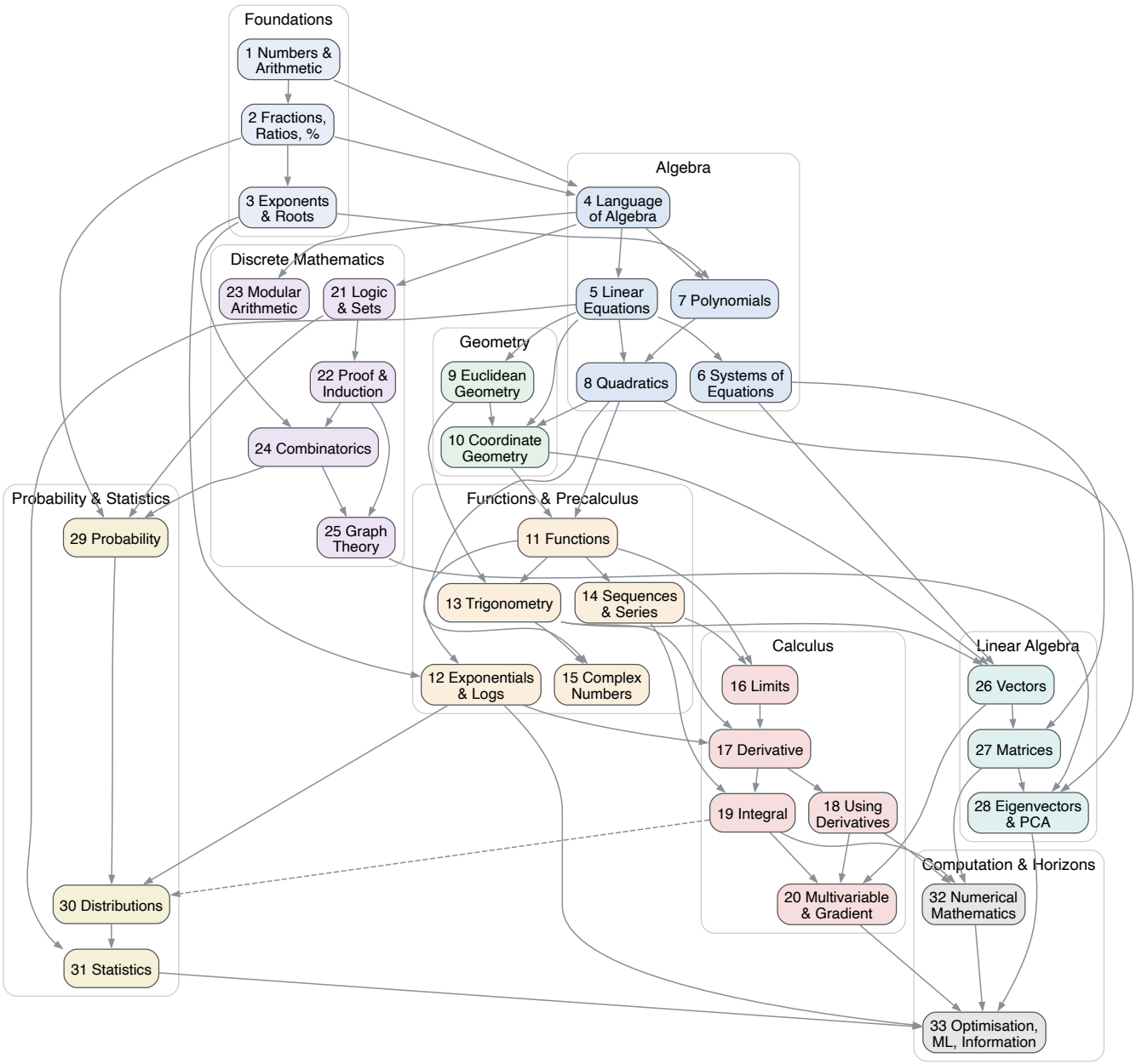


Figure 1. The knowledge map. Arrows point from prerequisite to dependent chapter.

The three volumes cut the map along its natural seams. Volume 1 is the foundation every other chapter rests on (Parts I–IV). Volume 2 holds the continuous branches: calculus, linear algebra and numerical mathematics. Volume 3 holds the discrete and probabilistic branches and ends with the Horizons capstone, which draws on both.

Computer-science connection: this map is a DAG

The map is a *directed acyclic graph*. "Find an order to study the chapters so every prerequisite comes first" is exactly **topological sorting**, the same problem a build system solves when it decides which files to compile first, or a package manager solves when it installs dependencies. You'll implement it in Volume 3, Chapter 25 . The numbering of this book is one valid topological order, but there are many others.

The spine and the branches

The **spine** is the classical route to calculus:

Arithmetic → Algebra → Geometry → Functions → Trigonometry → Precalculus → Calculus

For a computer scientist, several **branches** deserve equal standing:

Branch	Why it matters in computing
Calculus (Volume 2, Chapter 16 –Volume 2, Chapter 20)	Continuous change, optimisation, physics simulation, the training of machine-learning models.
Linear algebra (Volume 2, Chapter 26 –Volume 2, Chapter 28)	Graphics, transformations, embeddings, search, machine learning.
Probability (Volume 3, Chapter 29 , Volume 3, Chapter 30)	Uncertainty, randomised algorithms, hashing, machine learning.
Statistics (Volume 3, Chapter 31)	Inference, experimentation, A/B testing, metrics.
Logic and sets (Volume 3, Chapter 21)	Programming languages, databases, verification.
Proof (Volume 3, Chapter 22)	Algorithm correctness, loop invariants, reasoning about code.
Number theory (Volume 3, Chapter 23)	Hashing, checksums, cryptography.
Combinatorics (Volume 3, Chapter 24)	Algorithm analysis, probability, complexity.
Graph theory (Volume 3, Chapter 25)	Networks, search, compilers, dependency resolution, distributed systems.
Numerical mathematics (Volume 2, Chapter 32)	How computers actually approximate mathematical answers.

Volume 3, Chapter 33 brings the branches together in optimisation, machine learning and information theory.

Suggested routes

These are starting points, not prescriptions. Use the mastery checks to skip what you already know.

The calculus route

Chapter 1 → Chapter 2 → Chapter 3 → Chapter 4 → Chapter 5 → Chapter 7 → Chapter 8 → Chapter 10 → Chapter 11 → Chapter 12 → Chapter 13 → Chapter 14 → Volume 2, Chapter 16 → Volume 2, Chapter 17 → Volume 2, Chapter 18 → Volume 2, Chapter 19 .

The computer-science core

Foundations and Algebra, then Volume 3, Chapter 21 → Volume 3, Chapter 22 → Volume 3, Chapter 23 → Volume 3, Chapter 24 → Volume 3, Chapter 25 → Volume 3, Chapter 29 . These can run alongside the calculus route.

The machine-learning route

The calculus route through Volume 2, Chapter 18 , plus Volume 2, Chapter 26 → Volume 2, Chapter 27 → Volume 2, Chapter 28 , then Volume 3, Chapter 29 → Volume 3, Chapter 30 → Volume 3, Chapter 31 , then Volume 2, Chapter 20 → Volume 3, Chapter 33 .

The "helping with homework" route

Follow the US-curriculum sidebars. Appendix D maps each school course to the chapters that cover it.

From map to skill tree

Each chapter breaks down into skills. In a future companion app, the map becomes a skill tree whose progress reflects **demonstrated understanding**, not the number of problems ground through:

```
Algebra                          Geometry
  ● Linear equations               ● Coordinates
  ● Quadratics                     ○ Vectors
  ● Functions                      ○ Transformations
  ○ Exponents
  ○ Logarithms
```

Because the tree records dependencies, it can tell you more than "wrong answer". For example: *You understand equations but seem rusty on exponent laws. Refresh those two ideas before logarithms.* That's the purpose of the "Where this chapter sits" sidebar at the top of every chapter and the decision rule at the bottom.

US curriculum

A rough correspondence, detailed in Appendix D:

- **Elementary school**: Chapter 1, Chapter 2, early Chapter 9
- **Middle school / Pre-Algebra**: Chapter 2, Chapter 3, Chapter 4
- **Algebra I**: Chapter 5, Chapter 6, early Chapter 7, Chapter 8, Chapter 11
- **Geometry**: Chapter 9, Chapter 10
- **Algebra II**: Chapter 7, Chapter 12, Chapter 15, Chapter 14
- **Precalculus**: Chapter 11, Chapter 13, Chapter 14, introduction to Volume 2, Chapter 16
- **AP Calculus**: Volume 2, Chapter 16 to Volume 2, Chapter 19
- **AP Statistics**: Volume 3, Chapter 29 , Volume 3, Chapter 30 , Volume 3, Chapter 31
- **University (CS / math)**: discrete mathematics, linear algebra, multivariable calculus, numerical methods

Part I: Foundations

Chapter 1. Numbers, the Number Line, and Arithmetic

Chapter map: we begin where everything begins — with numbers. Not to relearn how to add, but to reconnect the arithmetic you can already do with the vocabulary, structure, and reasons behind it. We will see why number systems kept having to grow, how place value makes arithmetic (and computers) work, which laws of arithmetic quietly power all of algebra, why a negative times a negative must be positive, why "order of operations" is a convention rather than a law of nature, and how fixed-width integers in a computer behave differently from the integers of mathematics.

Why it matters: every quantity you will ever compute with (a bank balance, a sensor reading, a pixel coordinate, a 64-bit counter) lives in one of these number systems, and most arithmetic bugs, in spreadsheets and in production code alike, come from a value obeying different rules from the ones you assumed. This chapter is where those rules are made explicit.

Where this chapter sits

Prerequisites: none beyond everyday arithmetic. If you have read The Knowledge Map, you know this is the root of the dependency tree.

This chapter unlocks: fractions and decimals (Chapter 2), exponents (Chapter 3), algebraic expressions (Chapter 4), and — through divisibility and primes — number theory (Volume 3, Chapter 23). The arithmetic laws in Section 1.4 are the rules that every algebraic manipulation in the rest of the book relies on.

Time budget: if the mastery check at the end is easy, skim and move on. The sections most worth your time even if you are confident are Section 1.4, Section 1.5, and Section 1.9.

1.1. Kinds of number

1.1.1. The problem

Suppose you only had the counting numbers 1, 2, 3, … . You could count sheep, but you could not:

- say how many sheep are in an empty field;
- record that you *owe* someone three sheep;
- share two sheep fairly among three people;
- state the exact length of the diagonal of a square field whose sides are 1 kilometre.

Each of these is a perfectly ordinary question, and each one breaks the number system you had. The history of numbers is largely the history of patching those breaks. Each patch produced a larger system that contains the old one, and each larger system has a name and a symbol you will meet constantly in technical reading.

1.1.2. Intuition: each system fixes a failure of the last

Think of it as a sequence of feature requests.

- **Counting** gives the natural numbers. You can add and multiply them freely and always stay inside.
- **Subtraction** fails: $3 - 5$ has no answer among counting numbers. Fix: add zero and the negatives. Now you have the integers.
- **Division** fails: $2 \div 3$ has no answer among integers. Fix: allow ratios of integers. Now you have the rational numbers.
- **Measuring** fails: the diagonal of a unit square has length $\sqrt{2}$, and (as we will prove) no ratio of integers squares to exactly 2. Fix: fill every gap in the line. Now you have the real numbers.

Later in the book (Chapter 15) one more failure — $x^2 = -1$ has no real solution — will drive one more extension, to the complex numbers.

1.1.3. Vocabulary and notation

Before the vocabulary, one piece of notation. Mathematicians collect objects into **sets** and write $x \in S$ to mean "x is an element of the set S" (read "x is in S"). The symbol $\in$ is a stylised Greek epsilon, for the Latin *est*, "is". Writing $A \subset B$ means "every element of A is also an element of B" (read "A is a subset of B"). We will treat sets properly in Volume 3, Chapter 21 ; for now these two symbols are all you need.



Natural number

The **natural numbers** are the counting numbers $0, 1, 2, 3, \ldots$. The set of them is written $\mathbb{N}$ (a "blackboard bold" N). Whether $0 \in \mathbb{N}$ is a convention that varies: computer scientists and logicians almost always include 0; some number theorists start at 1. When it matters, authors write $\mathbb{N}_0$ (with zero) or $\mathbb{Z}^+$ (positive integers, without zero). This book includes 0.



Integer

An **integer** is a whole number that may be positive, negative, or zero: $\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots$. The set is written $\mathbb{Z}$, from the German *Zahlen*, "numbers". The integers are **closed** under addition, subtraction, and multiplication: doing any of those to two integers gives an integer.



Rational number

A **rational number** is any number that can be written as a ratio $\frac{p}{q}$ of integers p and q with $q \neq 0$. The set is written $\mathbb{Q}$, for *quotient*. Examples: $\frac{1}{2}$, $-\frac{7}{3}$, $5 = \frac{5}{1}$, $0.125 = \frac{1}{8}$. In decimal form a rational number either terminates or eventually repeats (shown in Chapter 2).

Irrational number

An **irrational number** is a real number that is *not* rational — it cannot be written as a ratio of integers. Its decimal expansion never terminates and never settles into a repeating block. Examples: $\sqrt{2} = 1.41421356\ldots$, $\pi = 3.14159265\ldots$, $e = 2.71828182\ldots$.

Real number

The **real numbers**, written $\mathbb{R}$, are all the rational and irrational numbers together: every point on the number line. Every length, every measurement, every point on a continuous scale is a real number. The defining extra property of $\mathbb{R}$ over $\mathbb{Q}$ is that it has *no gaps* (mathematicians call this **completeness**).

The systems nest inside one another:

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

Every natural number is an integer, every integer is rational ($n = n/1$), and every rational is real.

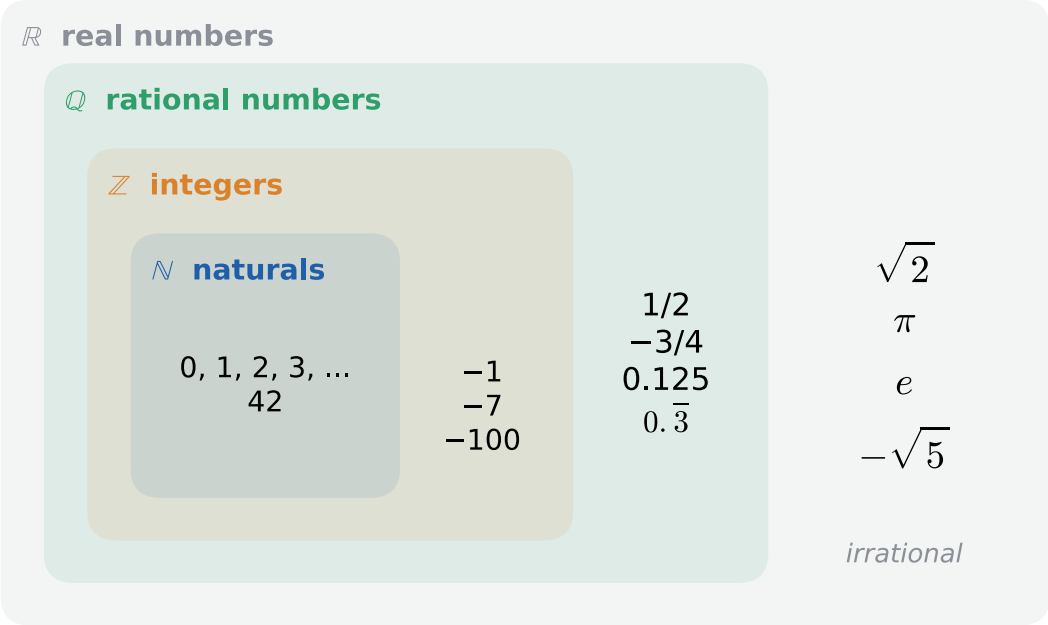


Figure 2. The number systems nest. Each is the previous one plus whatever was needed to repair a failure.

1.1.4. Why it works: $\sqrt{2}$ really is irrational

It is worth seeing **why** the rationals are not enough, because the argument is short, beautiful, and the template for a whole style of proof (proof by contradiction, Volume 3, Chapter 22).

Suppose, hoping for a contradiction, that $\sqrt{2} = p/q$ where p and q are integers and the fraction is already in lowest terms (no common factor — we can always cancel until this is true). Then

$$\begin{aligned} \sqrt{2} &= \frac{p}{q} \\ 2 &= \frac{p^2}{q^2} && \text{(square both sides)} \\ 2q^2 &= p^2. && \text{(multiply both sides by } q^2\text{)} \end{aligned}$$

So p^2 is even. The square of an odd number is odd, so p itself must be even: write $p = 2k$. Substituting,

$$\begin{aligned} 2q^2 &= (2k)^2 = 4k^2 \\ q^2 &= 2k^2. && \text{(divide both sides by 2)} \end{aligned}$$

So q^2 is even, and therefore q is even. But now p and q are both even, so they share the factor 2 — contradicting "lowest terms". The assumption was impossible: $\sqrt{2}$ is not a ratio of integers.



The scandal of the diagonal

Legend says the Pythagoreans, who believed "all is number" (meaning whole numbers and their ratios), were so disturbed by the irrationality of $\sqrt{2}$ that they drowned Hippasus of Metapontum for revealing it. The story is almost certainly embellished, but the mathematical shock was real: the Greeks responded by developing geometry largely separately from arithmetic for the next two thousand years. The word "irrational" originally meant simply "not a ratio", not "unreasonable".

1.1.5. Failure modes



Rational does not mean "has a short decimal"

$\frac{1}{3} = 0.333\ldots$ has an infinite decimal expansion but is perfectly rational. Irrationality is about the expansion **never repeating**, not about it being long. Conversely, a calculator display of 1.414213562 is a rational approximation — the calculator cannot store $\sqrt{2}$ exactly. Nor can your computer (Chapter 2 explains why even 0.1 is out of reach for binary floating point).

1.1.6. Where you'd use this

- **Money.** Prices look like decimals but are really integers in disguise: £19.99 is 1999 pence. Payment systems store whole pence or cents (integers, $\mathbb{Z}$) precisely so that totals add up exactly; storing money as `float` is a classic bug (Chapter 2).
- **Measurement.** Every physical length, mass, or time is a real number, but every *measurement* of it is a rational approximation. A bicycle wheel of diameter 0.7 m rolls $\pi \times 0.7 \approx 2.199$ m per turn; the exact distance is irrational, so an engineer decides how many digits matter (Section 1.10).
- **Counting things.** Stock levels, seats, packets, and array indices are natural numbers. A negative stock level or a fractional seat count is a sign that a value has escaped its proper number system, which makes "which set should this live in?" a useful validation rule for any database column.

Computer-science connection: number types are number systems

Programming languages mirror the mathematical hierarchy, imperfectly:

- `int` / `uint64` / `size_t` approximate $\mathbb{Z}$ and $\mathbb{N}$ — but with a fixed range (Section 1.9). Python's `int` is arbitrary precision and genuinely behaves like $\mathbb{Z}$, limited only by memory.
- Python's `fractions.Fraction`, or a pair of big integers, represents $\mathbb{Q}$ exactly.
- `float` / `double` are a finite set of rationals with power-of-two denominators, used to *approximate* $\mathbb{R}$. No computer type represents all of $\mathbb{R}$: there are uncountably many reals and only finitely many bit patterns.

When a bug report says "the totals don't add up", the root cause is very often a value that silently moved between these systems.

US curriculum

Whole numbers and integers are elementary and middle-school (grades 5–7) material. The classification natural / whole / integer / rational / irrational / real appears explicitly in **Grade 8** (Common Core 8.NS) and is revisited in **Algebra I**. Note: many US textbooks say **whole numbers** for $\{0, 1, 2, \ldots\}$ and **natural numbers** or **counting numbers** for $\{1, 2, 3, \ldots\}$.

1.2. The number line

1.2.1. Intuition

The single most useful picture in elementary mathematics is a straight line with a point marked 0, a point marked 1 to its right, and every real number occupying exactly one point. Positive numbers lie to the right of zero, negatives to the left, and the distance from zero records size.

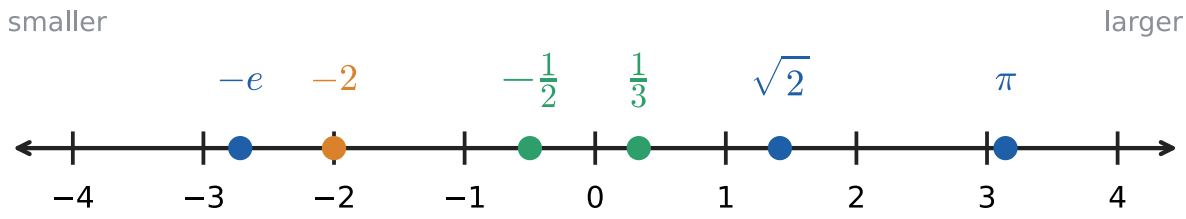


Figure 3. Every real number is a point on the line: integers, fractions, and irrationals alike.

The line turns arithmetic into motion. Adding 3 means stepping 3 units right. Subtracting 3 means stepping 3 units left. Multiplying by 2 means stretching everything away from zero by a factor of 2; multiplying by -1 means reflecting through zero. You will see these geometric readings return in vectors (Volume 2, Chapter 26) and complex numbers (Chapter 15), where multiplication becomes rotation.

1.2.2. Vocabulary and notation



Order and inequality symbols

$a < b$ means "a is less than b": a lies to the *left* of b on the number line. $a > b$ means "greater than". $a \le b$ means "less than or equal to", $a \ge b$ "greater than or equal to", and $a \ne b$ "not equal to". The narrow end of $<$ points at the smaller number. A number is **positive** if it is > 0, **negative** if < 0, and **non-negative** if ≥ 0; zero is neither positive nor negative.

Two facts about the line deserve to be said out loud, because they become important in calculus:

- **Density.** Between any two different rational numbers there is another one (their average). So there is no "next" rational after 0 — the rationals are packed infinitely tightly.
- **Gaps anyway.** Despite that density, the rationals miss points such as $\sqrt{2}$. The reals fill every gap. This "no gaps" property is exactly what makes limits (Volume 2, Chapter 16) work.



Negative numbers and "smaller"

$-10 < -2$, even though 10 is "bigger" than 2. On the number line -10 is further left. In everyday speech people say "a bigger loss" for -10, which describes *magnitude*, not order. When order matters, think "further left = less". Magnitude is captured by absolute value (Section 1.7).



Zooming the number line

A number line you can zoom indefinitely. Mark $\sqrt{2}$ and keep zooming: each zoom reveals another decimal digit, and the nearest rational points with small denominators ($\frac{7}{5}$, $\frac{17}{12}$, $\frac{41}{29}$, ...) light up and are left behind, but the point itself never lands on a rational.

Try it: Best fractions for √2

1.3. Place value and number bases

1.3.1. The problem

Write down the number of seconds in a year. You probably wrote something like 31536000, and you did not need a new symbol for it. Compare Roman numerals: XXXI million … there is no good way. The reason modern notation handles any size with only ten symbols is **place value**: the meaning of a digit depends on its position. Understanding place value precisely lets you work in any base — and base 2 and base 16 are the native tongues of computers.

1.3.2. Intuition

In 2047 the "2" means two thousands, the "0" means no hundreds, the "4" means four tens, and the "7" means seven ones. Each position is worth ten times the position to its right. Nothing is special about ten except that we have ten fingers. Make each position worth *two* times the one to its right, and you

have binary; sixteen, and you have hexadecimal.

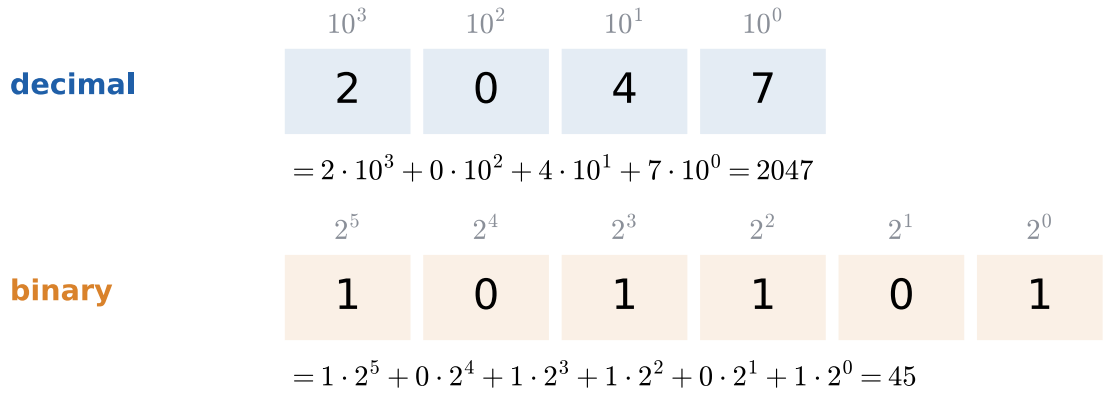


Figure 4. Positional notation: each position is worth a power of the base.

1.3.3. Why it works

In base b a digit string $d_k d_{k-1} \ldots d_1 d_0$ (each digit d_i between 0 and $b - 1$) stands for

$$d_k b^k + d_{k-1} b^{k-1} + \cdots + d_1 b^1 + d_0 b^0.$$

Here b^k means b multiplied by itself k times, and $b^0 = 1$ (powers are the subject of Chapter 3, which also explains why $b^0 = 1$). The subscripts on d_i are labels, not exponents: d_0 is "the digit in position 0", the rightmost one.

Every non-negative integer has exactly one such representation (without leading zeros). The reason is division with remainder: dividing by b peels off the last digit as the remainder and shifts the rest down one position as the quotient. Repeat until nothing is left.

1.3.4. Vocabulary and notation



Base, digit, and positional notation

The **base** (or **radix**) b of a positional system is how many times larger each position is than the one to its right. A **digit** is one of the symbols $0, 1, \ldots, b - 1$. **Positional** (or **place-value**) notation means a digit's value is the digit times b^{position}, counting positions from 0 at the right. When the base is not obvious we write it as a subscript: $101101_2 = 45_{10}$.



Binary, bit, byte, hexadecimal

Binary is base 2, with digits 0 and 1. A single binary digit is a **bit**; eight bits make a **byte**. **Hexadecimal** ("hex") is base 16, with digits **0–9** then A=10, B=11, C=12, D=13, E=14, F=15. Because $16 = 2^4$, each hex digit corresponds to exactly four bits (a **nibble**). **Octal** is base 8 (three bits per digit). Programming languages mark bases with prefixes: `0b101101`, `0x2D`, `0o55` all mean 45.

1.3.5. The method

Example 2. Procedure — convert a non-negative integer to base b

Input: a non-negative integer n and a base $b \geq 2$.

Steps:

1. If $n = 0$, the answer is "0". Stop.
2. Divide n by b, obtaining quotient q and remainder r (so $n = qb + r$ with $0 \leq r < b$).
3. Record r as the next digit, working from right to left.
4. Replace n by q. If $n > 0$, go back to step 2.

Output: the recorded digits, read in reverse order of discovery (last remainder is the leftmost digit).

Example 3. Procedure — convert a base-b digit string to an integer (Horner's rule)

Input: digits $d_k, d_{k-1}, \ldots, d_0$ (left to right) and base b.

Steps:

1. Set $v = 0$.
2. For each digit d, from left to right: set $v = v \cdot b + d$.

Output: v, the value.

Horner's rule avoids computing powers at all: each step "shifts left one place" by multiplying by b and then drops the new digit into the empty units position.

1.3.6. Worked examples

Example 1: 45 to binary. Divide repeatedly by 2.

$$\begin{aligned} 45 &= 22 \times 2 + 1 && \text{digit } d_0 = 1 \\ 22 &= 11 \times 2 + 0 && \text{digit } d_1 = 0 \\ 11 &= 5 \times 2 + 1 && \text{digit } d_2 = 1 \\ 5 &= 2 \times 2 + 1 && \text{digit } d_3 = 1 \\ 2 &= 1 \times 2 + 0 && \text{digit } d_4 = 0 \\ 1 &= 0 \times 2 + 1 && \text{digit } d_5 = 1 \text{ (quotient is 0: stop)} \end{aligned}$$

Reading the digits from d_5 down to d_0: $45_{10} = 101101_2$. Check by expanding:
$32 + 0 + 8 + 4 + 0 + 1 = 45$. ✓

Example 2: 2047 to hexadecimal.

$$\begin{aligned} 2047 &= 127 \times 16 + 15 && \text{digit F} \\ 127 &= 7 \times 16 + 15 && \text{digit F} \\ 7 &= 0 \times 16 + 7 && \text{digit 7 (stop)} \end{aligned}$$

So $2047_{10} = 7\mathrm{FF}_{16}$. Check: $7 \times 256 + 15 \times 16 + 15 = 1792 + 240 + 15 = 2047$. ✓

Example 3: $2\mathrm{D}_{16}$ to decimal by Horner's rule. Start with $v = 0$. Digit 2: $v = 0 \times 16 + 2 = 2$. Digit D (=13): $v = 2 \times 16 + 13 = 45$. So $2\mathrm{D}_{16} = 45$.

1.3.7. Alternative approach: grouping bits

Between binary and hex (or octal) there is a much faster route than going via decimal. Since one hex digit is exactly four bits, group the binary digits in fours *from the right* and translate each group:

$$101101_2 = \underbrace{0010}_{2}\,\underbrace{1101}_{\mathrm{D}} = 2\mathrm{D}_{16}.$$

(The underbrace labels what each group of four bits becomes.) Going the other way, expand each hex digit into four bits: $7\mathrm{FF}_{16} = 0111\ 1111\ 1111_2$.

The other common method for decimal-to-binary is **subtracting powers of two**: find the largest power of two not exceeding n, write a 1 there, subtract, repeat. For 45: 32 fits (remainder 13), 16 does not, 8 fits (remainder 5), 4 fits (remainder 1), 2 does not, 1 fits. Digits 101101 again.

1.3.8. Choosing a method

- Binary ↔ hex or octal: **always group bits**. It is mechanical and nearly error-free.
- Decimal → binary for small numbers you can see the powers of 2 in (anything under about 1000): **subtract powers of two** — it is the fastest mental method.
- Decimal → any base for larger numbers, or in code: **repeated division**.
- Any base → decimal: **Horner's rule**, which is also how every `parseInt` is written.

1.3.9. Implementation

```python
DIGITS = "0123456789ABCDEF"

def to_base(n, b):
    """Return the base-b digit string of a non-negative integer n (2 <= b <= 16)."""
    if n == 0:
        return "0"
    digits = []
    while n > 0:
        n, r = divmod(n, b)        # quotient and remainder in one step
        digits.append(DIGITS[r])   # remainders come out least-significant first
    return "".join(reversed(digits))

def from_base(s, b):
    """Inverse: evaluate a digit string in base b (Horner's rule)."""
    value = 0
    for ch in s:
        value = value * b + DIGITS.index(ch.upper())
    return value

print(to_base(45, 2), to_base(2047, 16), to_base(255, 2))
print(from_base("101101", 2), from_base("7FF", 16), int("7FF", 16))
```

```
101101 7FF 11111111
45 2047 2047
```

Python's built-ins `bin`, `hex`, `oct`, `int(s, base)`, and format strings such as `f"{45:08b}"` do the same jobs; the point of writing them yourself once is to see that they are just the two procedures above.



> *A leading zero that changes the number*
>
> In C, C++, Java, and JavaScript (non-strict mode), an integer literal with a leading zero is **octal**: `010` is eight, not ten, and `09` is an error. Zip codes and phone numbers stored as integers have been mangled this way for decades. Python 3 forbids the ambiguous form and requires `0o10`. Rule of thumb: identifiers that happen to be made of digits (zip codes, account numbers) are strings, not numbers.

Computer-science connection: why hex is everywhere

Memory addresses, colour codes (`#1F5FA8` is red $\mathbf{1F}_{16} = 31$, green $\mathbf{5F}_{16} = 95$, blue $\mathbf{A8}_{16} = 168$), UUIDs, hashes, and machine code are all written in hex. Not because hex is meaningful to the machine — the machine only has bits — but because hex is the most compact human-readable notation that maps **exactly** onto bits: one byte is always two hex digits, so you can read bit patterns off at a glance. Decimal does not have this property: the decimal digits of a number do not line up with its bytes.



Base 60 is still on your wrist

The Babylonians used base 60 around 2000 BCE, with a place-value system but (for a long time) no zero symbol, so they had to infer from context whether a gap meant "nothing in this place". We still inherit their base in 60 seconds per minute, 60 minutes per hour, and 360 degrees in a circle. Sixty is convenient because it divides evenly by 2, 3, 4, 5, 6, 10, 12, 15, 20, and 30. A true zero digit, acting as a placeholder, was developed in India by about the 7th century CE and reached Europe through Arabic mathematicians — hence "Hindu–Arabic numerals".



Base explorer

Type a number and watch it displayed simultaneously in bases 2, 8, 10, and 16, with each digit's place value shown as a stack of blocks. Click a bit to toggle it and see the decimal and hex values update; notice that toggling bit k always changes the value by exactly 2^k.

Try it: 255 in every base

1.3.10. Exercises

1. Write 1011_2, 777_8, and FF_{16} in decimal.
2. Convert 100 to binary by repeated division, then convert your answer to hex by grouping bits.
3. A colour is stored as `0x00FF7F`. What are its red, green, and blue components in decimal?
4. Explain, using the place-value formula, why appending a 0 to the right of a binary numeral doubles its value. What does appending a 0 do in base b?

▼ **Answers**

1. $1011_2 = 8 + 0 + 2 + 1 = 11$. $777_8 = 7 \cdot 64 + 7 \cdot 8 + 7 = 448 + 56 + 7 = 511$. $\mathrm{FF}_{16} = 15 \cdot 16 + 15 = 255$.
2. $100 = 50 \cdot 2 + 0$, $50 = 25 \cdot 2 + 0$, $25 = 12 \cdot 2 + 1$, $12 = 6 \cdot 2 + 0$, $6 = 3 \cdot 2 + 0$, $3 = 1 \cdot 2 + 1$, $1 = 0 \cdot 2 + 1$. Reading remainders upward: 1100100_2. Grouping from the right: $0110\ 0100 = 64_{16}$. Check: $6 \cdot 16 + 4 = 100$. ✓
3. Red $00 = 0$, green $\mathrm{FF} = 255$, blue $7\mathrm{F} = 7 \cdot 16 + 15 = 127$. (A spring green.)
4. Appending a digit moves every existing digit up one position, multiplying its place value by the base, and puts 0 in the units place. So in base 2 the value doubles; in base b it is multiplied by b. This is why a left shift `x << 1` multiplies by 2.

US curriculum

Base-ten place value is a core **elementary** topic (grades 1–5, Common Core NBT strand). Other bases are rarely taught in standard US courses today; they appear in enrichment, in some **Pre-Algebra** texts, and in **AP Computer Science Principles**, which covers binary explicitly.

1.4. The four operations and their laws

1.4.1. The problem

How do you work out 17×25 in your head? Many people do something like "$17 \times 100 = 1700$, divide by 4, that's 425." Or "$17 \times 25 = 10 \times 25 + 7 \times 25 = 250 + 175 = 425$." Both are correct. But *why* are you allowed to do that? Neither is the long-multiplication algorithm you were taught.

The answer is a handful of laws that addition and multiplication obey. They look so obvious that it is easy to skip past them — but they are the entire rulebook of algebra. Every time you "simplify", "expand", "factor", "collect like terms", or "move something to the other side", you are applying one of these laws. Knowing them by name means you can justify every step you take, and spot the step that is not justified.

1.4.2. Vocabulary and notation



Sum, difference, product, quotient

The result of an addition is a **sum**; its inputs are **terms** (or **addends**). The result of a subtraction is a **difference**. The result of a multiplication is a **product**; its inputs are **factors**. The result of a division is a **quotient**: in $a \div b$, a is the **dividend** and b the **divisor**. Multiplication may be written $a \times b$, $a \cdot b$, or simply ab when at least one factor is a letter.



Commutative, associative, distributive

An operation is **commutative** if order of the inputs does not matter: $a + b = b + a$, $ab = ba$. It is **associative** if grouping does not matter: $(a + b) + c = a + (b + c)$, $(ab)c = a(bc)$. Multiplication **distributes** over addition: $a(b + c) = ab + ac$. Mnemonic: *commute* = move around; *associate* = who you group with; *distribute* = hand out to each.



Identity and inverse

An **identity element** for an operation leaves things unchanged: $a + 0 = a$ (0 is the **additive identity**), $a \times 1 = a$ (1 is the **multiplicative identity**). An **inverse** undoes: $-a$ is the **additive inverse** (or **opposite**) of a because $a + (-a) = 0$; $\frac{1}{a}$ is the **multiplicative inverse** (or **reciprocal**) of $a \neq 0$ because $a \cdot \frac{1}{a} = 1$.

Collected in one place, for all real numbers a, b, c:

Law	Addition	Multiplication
Commutative	$a + b = b + a$	$ab = ba$
Associative	$(a + b) + c = a + (b + c)$	$(ab)c = a(bc)$
Identity	$a + 0 = a$	$a \cdot 1 = a$
Inverse	$a + (-a) = 0$	$a \cdot \frac{1}{a} = 1 \quad (a \neq 0)$
Distributive	(links $+$ and $\times$)	$a(b + c) = ab + ac$

1.4.3. Intuition

The distributive law is the one that does the heavy lifting, and it has a lovely picture: a rectangle of height a and width $b + c$ has the same area as two rectangles of widths b and c placed side by side.

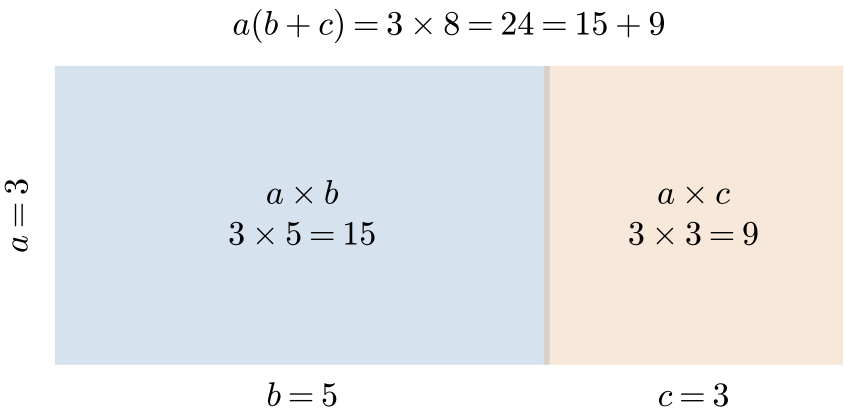


Figure 5. The distributive law as area: $a(b + c)$ is the whole rectangle; $ab + ac$ is the sum of its two parts.

Commutativity of multiplication has a picture too: a grid of 3 rows of 5 dots is the same grid as 5 columns of 3 dots, just turned sideways.

1.4.4. What about subtraction and division?

Subtraction and division are **not** commutative ($5 - 3 \neq 3 - 5$, $6 \div 2 \neq 2 \div 6$) and **not** associative:

$$(8 - 4) - 2 = 2 \quad \text{but} \quad 8 - (4 - 2) = 6.$$

The clean way to think about this — and the way algebra will treat it — is that subtraction and division are not really separate operations. They are addition and multiplication by an inverse:

$$a - b = a + (-b), \qquad a \div b = a \cdot \frac{1}{b}.$$

Rewrite every subtraction as "add the opposite" and every division as "multiply by the reciprocal", and you are back in the world where everything commutes and associates. $8 - 4 - 2$ becomes $8 + (-4) + (-2)$, which you can group and reorder however you like.

1.4.5. Why these laws matter later

When you solve $3x + 5 = 20$ in Chapter 5, the steps are:

$$
\begin{aligned}
3x + 5 &= 20 && \\
3x + 5 + (-5) &= 20 + (-5) && \text{(add the additive inverse of 5 to both sides)} \\
3x + 0 &= 15 && \text{(inverse: } 5 + (-5) = 0\text{)} \\
3x &= 15 && \text{(identity: } 3x + 0 = 3x\text{)} \\
\tfrac{1}{3}(3x) &= \tfrac{1}{3} \cdot 15 && \text{(multiply both sides by the reciprocal of 3)} \\
(\tfrac{1}{3} \cdot 3)x &= 5 && \text{(associative law)} \\
1 \cdot x &= 5 && \text{(inverse)} \\
x &= 5 && \text{(identity)}
\end{aligned}
$$

You would never write it out this laboriously in practice. But every "move it to the other side" is secretly this chain. When you meet matrices (Volume 2, Chapter 27), multiplication will **stop** being commutative ($AB \neq BA$ in general), and suddenly it matters a great deal which laws you were relying on. Mathematicians give names to structures by which laws they obey: a set with $+$ and $\times$ obeying all of the laws above is called a **field**. $\mathbb{Q}$ and $\mathbb{R}$ are fields; $\mathbb{Z}$ is not (2 has no integer reciprocal).

1.4.6. Worked example: mental arithmetic as applied algebra

Compute 98×7 mentally, justifying each step.

$$
\begin{aligned}
98 \times 7 &= (100 - 2) \times 7 && \text{(rewrite 98)} \\
&= 100 \times 7 - 2 \times 7 && \text{(distributive law)} \\
&= 700 - 14 && \\
&= 686. &&
\end{aligned}
$$

Compute $4 \times 37 \times 25$.

$$
\begin{aligned}
4 \times 37 \times 25 &= 37 \times 4 \times 25 && \text{(commutative law)} \\
&= 37 \times (4 \times 25) && \text{(associative law)} \\
&= 37 \times 100 = 3700. &&
\end{aligned}
$$

The second example shows why these laws are really about **freedom**: you are allowed to choose the order that makes the work easy.

1.4.7. Where you'd use this

- **Shopping and tipping.** Four items at £2.99: $4 \times 2.99 = 4 \times (3 - 0.01) = 12 - 0.04 = 11.96$ (distributive law). A 15% tip on a \$46 bill: 10% is \$4.60 and 5% is half of that, \$2.30, so the tip is \$6.90, because $0.15 \times 46 = (0.10 + 0.05) \times 46$.
- **Mechanics.** A constant force of 30 N pushes a crate 4 m and then a further 6 m. The work done is $30 \times 4 + 30 \times 6 = 30 \times (4 + 6) = 300$ J. Factoring out the common force is the distributive law, and it is why physicists can add displacements first and multiply once.
- **Compilers and databases.** An optimising compiler rewrites `x*8 + x*2` as `x*10` (distributive law) and `(x*4)*25` as `x*100` (associative law). A database evaluating `SUM(price * 1.2)` may compute `1.2 * SUM(price)` instead: one multiplication instead of a million. Both rewrites are legal for integers and real numbers, and only approximately legal for floats, which is exactly the subject of the sidebar below.

1.4.8. Division by zero



Why you cannot divide by zero

"$a \div b = c$" means "c is the number with $c \times b = a$". Try $6 \div 0 = c$: we need $c \times 0 = 6$, but anything times 0 is 0, so no c works. Try $0 \div 0 = c$: we need $c \times 0 = 0$, and *every* c works, so there is no single answer. Either way, division by zero has no sensible value, so it is left **undefined**. Zero is the one real number with no reciprocal. Many algebra "proofs" that $1 = 2$ hide a division by an expression that equals zero.

Computer-science connection: floating-point addition is not associative

The laws above hold for real numbers. They do **not** all hold for the floating-point numbers in your computer, because every operation rounds its result:

```python
>>> (0.1 + 0.2) + 0.3
0.6000000000000001
>>> 0.1 + (0.2 + 0.3)
0.6
>>> 1e16 + 1 - 1e16
0.0
```

This matters more than it looks. A parallel sum (map-reduce, GPU reduction, multithreaded `sum`) adds numbers in an order that depends on scheduling, so the same data can give slightly different totals on different runs. Compilers are forbidden to reorder floating-point arithmetic unless you pass flags like `-ffast-math`, precisely because associativity fails. Conversely, **string concatenation** is associative but not commutative, and **integer** addition in a fixed-width register is associative and commutative even when it overflows (Section 1.9) — which is one reason wraparound arithmetic is so convenient for hardware.

Try it: Sum in different orders .



Why "distributive"?

The names "commutative" and "distributive" were coined by the French mathematician François-Joseph Servois in 1814; "associative" came from William Rowan Hamilton in the 1840s, who needed the word because he had just invented the quaternions, a number system in which multiplication is associative but **not** commutative. Until people built systems where the laws failed, nobody had needed names for them.

Laws as tiles

Drag the dividing line in a $a \times (b + c)$ rectangle and watch the two partial areas and their sum update. Then switch to "floating-point mode": generate a random list of 1,000 floats, sum it in forward, reverse, and shuffled order, and display the differing totals.

Try it: Floating-point mode: summation order · Floating-point mode: sum order

1.4.9. Exercises

1. Name the law used in each step: (a) $3 + x = x + 3$; (b) $2(x + 5) = 2x + 10$; (c) $(5 \cdot x) \cdot 2 = 5 \cdot (x \cdot 2)$.
2. Compute 999×6 and 25×48 mentally, writing down the law behind each step.
3. Show by example that division is not associative.
4. Expand $(a + b)(c + d)$ using only the distributive law (twice), and draw the matching area picture.

▼ **Answers**

1. (a) Commutative law of addition. (b) Distributive law. (c) Associative law of multiplication.
2. $999 \times 6 = (1000 - 1) \times 6 = 6000 - 6 = 5994$ (distributive).
   $25 \times 48 = 25 \times (4 \times 12) = (25 \times 4) \times 12 = 100 \times 12 = 1200$ (associative).
3. $(16 \div 4) \div 2 = 4 \div 2 = 2$, but $16 \div (4 \div 2) = 16 \div 2 = 8$.
4. Treat $(a + b)$ as a single number and distribute it over $c + d$: $(a + b)(c + d) = (a + b)c + (a + b)d$. Distribute again (with commutativity to put the factor on the left): $= ac + bc + ad + bd$. The picture is a rectangle of width $a + b$ and height $c + d$ cut into four smaller rectangles. This is the origin of "FOIL", which Chapter 7 generalises.

US curriculum

The properties of operations are named and used from **grade 3** onward (Common Core 3.OA.5), and formalised in **Pre-Algebra** and **Algebra I**, where students justify equation-solving steps by citing them.

1.5. Negative numbers

1.5.1. The problem

A bank account holds \$40 and you write a cheque for \$70. The balance is -30. The temperature is $-5°\text{C}$ and falls by 8 degrees: it is now $-13°\text{C}$. These are easy. But what is $(-3) \times (-4)$? School says 12 — "two negatives make a positive" — but if you were never told *why*, the rule feels arbitrary, and arbitrary rules are the ones that get misremembered.

1.5.2. Intuition: adding and subtracting on the line

On the number line, a negative number is a step to the left.

- $5 + (-8)$: start at 5, step 8 left, land on -3.
- $5 - 8$: the same thing, since $a - b = a + (-b)$.
- $5 - (-8)$: "subtracting a step to the left" — undoing a debt — is a step to the right: $5 - (-8) = 5 + 8 = 13$.

The last one is just the observation that the opposite of the opposite is the original: $-(-8) = 8$, because 8 is the number that, added to -8, gives 0.

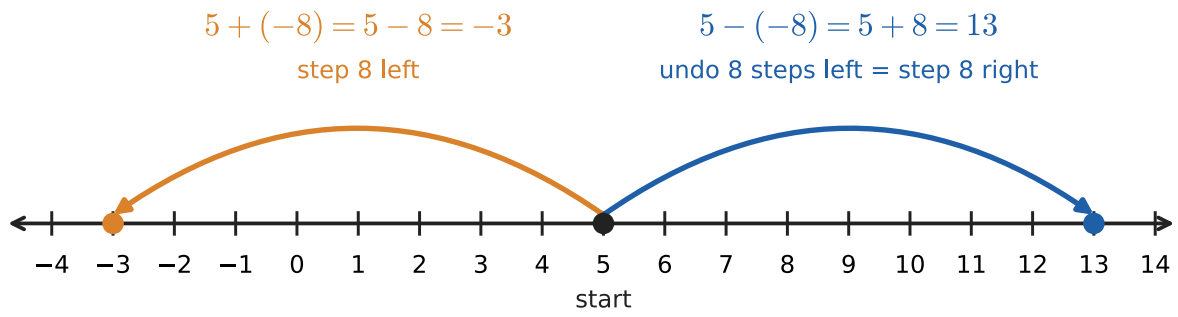


Figure 6. Signed numbers as hops: adding -8 steps 8 left; subtracting -8 undoes a step left, so it steps 8 right.

1.5.3. Why negative times negative is positive

There are two good arguments. The first is a **pattern** argument that convinces; the second is an **algebraic** argument that proves.

Pattern. Multiply -2 by a factor that decreases by 1 each line:

$$\begin{aligned} 3 \times (-2) &= -6 \\ 2 \times (-2) &= -4 \\ 1 \times (-2) &= -2 \\ 0 \times (-2) &= 0 \\ (-1) \times (-2) &= ? \\ (-2) \times (-2) &= ? \end{aligned}$$

The first four lines are uncontroversial ($3 \times (-2)$ is three debts of 2). Each time the first factor drops by 1, the product *rises* by 2. If the pattern continues — and anything else would put a kink in a straight line — then $(-1) \times (-2) = 2$ and $(-2) \times (-2) = 4$.

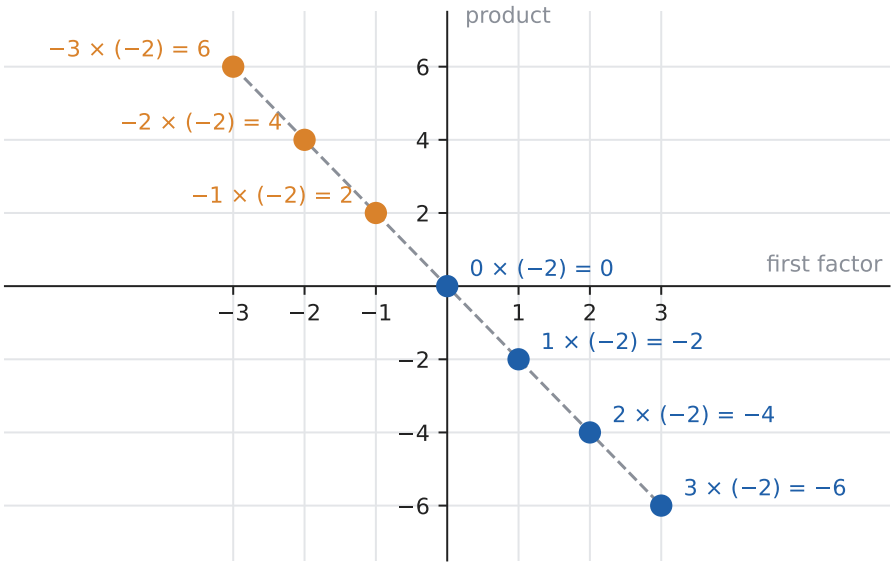


Figure 7. Multiplying by -2: the products lie on a straight line. Continuing it to the left forces negative × negative to be positive.

Proof from the laws. The pattern is persuasive, but here is the real reason: *if we want the distributive law to keep working for negative numbers, we have no choice.* We use only the laws from Section 1.4.

First, anything times zero is zero. For any number a:

$$\begin{aligned} a \cdot 0 &= a \cdot (0 + 0) && \text{(0 is the additive identity: } 0 + 0 = 0\text{)} \\ a \cdot 0 &= a \cdot 0 + a \cdot 0 && \text{(distributive law)} \\ 0 &= a \cdot 0 && \text{(add } -(a \cdot 0) \text{ to both sides)} \end{aligned}$$

Now the main event. Start from something we know is zero and expand it two ways:

$$\begin{aligned} (-1) \cdot \big(1 + (-1)\big) &= (-1) \cdot 0 = 0 && \text{(since } 1 + (-1) = 0\text{)} \\ (-1) \cdot \big(1 + (-1)\big) &= (-1) \cdot 1 + (-1)(-1) && \text{(distributive law)} \\ &= -1 + (-1)(-1). && \text{(1 is the multiplicative identity)} \end{aligned}$$

Putting the two lines together, $-1 + (-1)(-1) = 0$. So $(-1)(-1)$ is the number that, added to -1, gives zero. That number is 1. Hence

$$(-1)(-1) = 1.$$

The general case follows by writing $-a = (-1)a$: $(-a)(-b) = (-1)(-1)ab = ab$.

The moral: "negative times negative is positive" is not a rule someone chose. It is **forced** by the distributive law. Any other answer would break the arithmetic that lets you expand brackets.

1.5.4. The method

Example 4. Procedure — signs in multiplication and division

Input: a product or quotient of signed numbers.

Steps:

1. Ignore the signs and compute with the sizes (absolute values).
2. Count the negative factors (dividing by a negative counts as a negative factor).
3. If the count is even, the result is positive; if odd, negative.

Output: the signed result.

Worked example. $\dfrac{(-6)\times 4\times(-5)}{-3}$. Sizes: $6\times 4\times 5/3 = 120/3 = 40$. Negatives: -6, -5, -3 — three, odd. Result: -40.

1.5.5. Where you'd use this

- **Everyday.** The temperature rises from $-5°\text{C}$ overnight to $3°\text{C}$ at noon: the change is $3-(-5)=8$ degrees. A golfer at -3 (three under par) who scores $+2$ on the next hole is at -1.
- **Physics.** In mechanics a sign encodes a direction. Take "up" as positive and throw a ball upward at 12 m/s; gravity gives an acceleration of $-9.8\ \text{m/s}^2$. After 2 seconds the velocity is $12+(-9.8)(2)=-7.6$ m/s: the negative sign tells you the ball is now falling. Charge, current direction, and profit/loss work the same way.
- **Computing.** Screen coordinates usually have y increasing *downward*, so "move up 10 pixels" is $\Delta y=-10$. Python's `a[-1]` is the last element of a list, `a[-2]` the one before. File seeks, timezone offsets (UTC -5), and version diffs are all signed quantities.

1.5.6. Failure modes



> ***-3^2 is not $(-3)^2$***
> By the standard convention (Section 1.6), exponents bind more tightly than the leading minus sign. So -3^2 means $-(3^2)=-9$, while $(-3)^2=9$. Python agrees: `-3**2` is `-9`. Spreadsheets (Excel, Google Sheets) *disagree* and evaluate `=-3^2` as 9. When the sign matters, add brackets.

Try it: –3² in five conventions .



Two minus signs that mean different things

The symbol "−" does two jobs. In $7 - 3$ it is **binary** subtraction (two inputs). In -3 or $-x$ it is **unary** negation ("the opposite of"). $-x$ is not necessarily a negative number: if $x = -5$ then $-x = 5$. Read $-x$ as "the opposite of x", never as "negative x", and many sign errors disappear.



A latecomer

Negative numbers were used routinely in China (red rods for positive, black for negative) and in India by the 7th century, where Brahmagupta gave the sign rules explicitly in terms of "fortunes" and "debts". European mathematicians resisted them for over a thousand years longer: as late as the 18th century, some respected authors called them "absurd" or "fictitious". Accountants, who had been using red ink for debts all along, were ahead of the mathematicians.

1.5.7. Exercises

1. Compute: (a) $-7 + 12$; (b) $-7 - 12$; (c) $-7 - (-12)$; (d) $(-7)(-12)$.
2. Compute $(-1)^5$, $(-1)^6$, and $(-2)^3 \times (-1)^2$.
3. Using the same style as the proof above, show that $(-1) \cdot a = -a$ for any a. (Hint: show $a + (-1)a = 0$.)

▼ **Answers**

1. (a) 5. (b) -19. (c) $-7 + 12 = 5$. (d) 84.
2. Five negative factors (odd): -1. Six (even): 1. $(-2)^3 = -8$, $(-1)^2 = 1$, product -8.
3. $a + (-1)a = 1 \cdot a + (-1)a = (1 + (-1))a = 0 \cdot a = 0$ (identity, distributive, inverse, and the "times zero" fact). So $(-1)a$ is a number which added to a gives 0 — by definition, that is $-a$.

US curriculum

Operations with negative numbers (integers and rational numbers) are a **grade 6–7** topic (Common Core 6.NS, 7.NS). Grade 7 standards explicitly ask students to understand that $(-1)(-1) = 1$ follows from the distributive property.

1.6. Order of operations

1.6.1. The problem

Every few months an expression such as

$$8 \div 2(2 + 2)$$

goes viral, with half the internet insisting the answer is 16 and the other half insisting it is 1. Who is right? The uncomfortable truth is that the question has no mathematical answer. Order of operations is a **convention** — an agreement about how to read a string of symbols — and this expression sits exactly where the conventions are ambiguous.

1.6.2. Intuition

A string like $2 + 3 \times 4$ could mean $(2 + 3) \times 4 = 20$ or $2 + (3 \times 4) = 14$. Arithmetic itself does not care; it is the *notation* that needs a rule. The convention that multiplication goes first was adopted because it makes the most common expressions short: polynomials like $3x^2 + 2x + 1$ would otherwise need brackets everywhere.

1.6.3. The convention



Order of operations (precedence)

Precedence says which operations are done first. The standard order is:

1. **Brackets** (parentheses) — innermost first;
2. **Exponents** (powers and roots);
3. **Multiplication and division** — equal rank, left to right;
4. **Addition and subtraction** — equal rank, left to right.

US mnemonic: **PEMDAS** ("Please Excuse My Dear Aunt Sally"). UK: **BODMAS** or **BIDMAS** (Brackets, Orders/Indices, Division, Multiplication, Addition, Subtraction). They are the same rule.



Associativity of operators (left/right)

When operators of equal rank appear in a row, **operator associativity** decides grouping. Subtraction and division are **left-associative**: $a - b - c = (a - b) - c$. Exponentiation is **right-associative**: $a^{b^c} = a^{(b^c)}$, so $2^{3^2} = 2^9 = 512$, not $8^2 = 64$. (This is a different meaning of "associative" from the law in Section 1.4: here it is a parsing rule for notation.)

1.6.4. The method

Example 5. Procedure — evaluate an arithmetic expression by hand

Input: an expression with numbers, brackets, and operators.

Steps:

1. Find an innermost pair of brackets and evaluate inside it using steps 2–4. Replace the bracket by its value. Repeat until no brackets remain.
2. Evaluate all exponents, right to left when stacked.
3. Sweep left to right, performing each multiplication or division as you meet it.
4. Sweep left to right, performing each addition or subtraction as you meet it.

Output: a single number.

1.6.5. Worked example

Evaluate $20 - 3 \times (4 - 1)^2 \div 9 + 2$.

$$\begin{aligned} 20 - 3 \times (4-1)^2 \div 9 + 2 &= 20 - 3 \times 3^2 \div 9 + 2 && \text{(brackets)} \\ &= 20 - 3 \times 9 \div 9 + 2 && \text{(exponent)} \\ &= 20 - 27 \div 9 + 2 && \text{(multiply: leftmost of } \times, \div) \\ &= 20 - 3 + 2 && \text{(divide)} \\ &= 17 + 2 && \text{(subtract: leftmost of } +, -) \\ &= 19. && \text{(add)} \end{aligned}$$

A common mistake is to do all additions before subtractions: $20 - (3 + 2) = 15$. Wrong — they have equal rank and go left to right.

1.6.6. Alternative approach: the expression tree

Precedence rules are really instructions for building a **tree**: each operator is a node whose children are its operands. $2 + 3 \times 4$ is a $+$ node with children 2 and (a $\times$ node with children 3 and 4). Evaluate the tree from the leaves upward. Once you see expressions as trees, precedence stops being a list to memorise and becomes a question of "which operator is at the root?" Chapter 4 develops this view.

1.6.7. Back to the viral problem

In $8 \div 2(2 + 2)$, strict PEMDAS gives $8 \div 2 \times 4 = 4 \times 4 = 16$. But many mathematicians, physicists, and textbooks treat **implied multiplication** (juxtaposition, writing $2(\ldots)$ or $2x$ with no symbol) as binding more tightly than an explicit $\div$, so $1/2x$ means $1/(2x)$. Under that reading the answer is $8 \div 8 = 1$. Neither convention is wrong; the expression is simply badly written. Real mathematics avoids $\div$ entirely after primary school and uses fraction bars, which make grouping visible: $\frac{8}{2}(2 + 2)$ or $\dfrac{8}{2(2+2)}$. *Try it:* Parse 8÷2(2+2) five ways .



Choosing how to write it

If an expression's meaning depends on a precedence rule that a reasonable reader might not share, add brackets. This applies doubly to code. Brackets are free; misreadings are not.

1.6.8. Where you'd use this

- **Spreadsheets.** To average three test scores in cells holding 70, 80, and 90, typing `=A1+B1+C1/3` gives $70 + 80 + 90/3 = 180$, not 80. The fix is `=(A1+B1+C1)/3`. This exact bug turns up in real gradebooks and budgets. *Try it:* See the gradebook bug's tree.
- **Physics formulas.** Kinetic energy is $\frac{1}{2}mv^2$: the exponent applies to v alone. For a 2 kg ball at 3 m/s that is $\frac{1}{2} \times 2 \times 3^2 = 9$ J. Squaring mv instead gives $\frac{1}{2}(2 \times 3)^2 = 18$ J, which is twice as large and wrong.
- **Finance.** Compound interest $P(1 + r)^n$ needs its brackets: the exponent must apply to the whole growth factor $1 + r$, not to r alone (Chapter 3).

Computer-science connection: operator precedence in languages

Every language has a precedence table, and they differ in exactly the places where human conventions are fuzzy:

- Python: `-2**2 == -4` (power binds tighter than unary minus), `2**3**2 == 512` (right-associative), `-7 // 2 == -4` (floor division rounds toward $-\infty$).
- C, C++, Java: `-7 / 2 == -3` (truncates toward zero). And the bitwise operators `&`, `|`, `^` have **lower** precedence than `==`, a historical accident Dennis Ritchie later regretted: `x & 1 == 0` parses as `x & (1 == 0)`.
- In C, `^` is XOR, not power — `2 ^ 3` is 1.
- Excel: `=-3^2` is 9 (unary minus first), the opposite of Python.

Parsers implement precedence with techniques such as **recursive descent** (one function per precedence level) or Dijkstra's **shunting-yard** algorithm, which converts infix notation to **Reverse Polish Notation** — postfix, as on old HP calculators: `3 4 2 * +`. RPN needs no precedence rules and no brackets at all, because the order is explicit.

Try it: $2^{3}2$: 512 or 64?; Excel's =-3^2 versus Python.

Precedence playground

Type an expression and see its parse tree drawn, with a selector for convention: "strict PEMDAS", "implied multiplication binds tighter", "Python", "C", "Excel". Watch the tree — and the answer — change for $8 \div 2(2+2)$ and -3^2.

Try it: Open the Precedence playground

1.6.9. Exercises

1. Evaluate $6 + 18 \div 3 \times 2 - 4$.
2. Evaluate $(2+3)^2 - 2 + 3^2$.
3. Without brackets, $2 \times 3 + 4^2 - 1 = 21$. Insert one pair of brackets to make the result (a) 97, (b) 36, (c) 10.
4. What does Python print for `2 ** -1`, `-2 ** 0.5` and `(-8) // 3`? Explain each.

▼ **Answers**

1. $18 \div 3 = 6$, then $6 \times 2 = 12$; so $6 + 12 - 4 = 14$.
2. $5^2 - 2 + 9 = 25 - 2 + 9 = 32$.
3. (a) $2 \times (3+4)^2 - 1 = 2 \times 49 - 1 = 97$. (b) $2 \times (3 + 4^2 - 1) = 2 \times 18 = 36$. (c) $2 \times 3 + 4^{(2-1)} = 6 + 4 = 10$ (or $(2 \times 3 + 4)^{(2-1)} = 10$). Brackets change which operation sits at the root of the expression tree; the same symbols can produce very different values.
4. `0.5`: $2^{-1} = \frac{1}{2}$ (Chapter 3). `-1.4142135623730951`: parsed as $-(2^{0.5})$. `-3`: floor of $-2.666\ldots$ is -3 (Python floors toward $-\infty$).

US curriculum

Order of operations is taught in **grades 5–6** (Common Core 5.OA.1, 6.EE.2c) with the PEMDAS mnemonic, and used constantly in **Pre-Algebra** and **Algebra I**.

1.7. Absolute value

1.7.1. The problem

Two sensors read -3.2 and 1.8. How far apart are their readings? A stock moved by -4%; how big was the move? We often need the *size* of a number without its sign, or the *distance* between two numbers without caring which is larger.

1.7.2. Intuition and definition

The **absolute value** of a number is its distance from zero on the number line. Distance is never negative.

Absolute value

The **absolute value** (or **magnitude**, or **modulus**) of x, written $|x|$, is its distance from 0:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x < 0. \end{cases}$$

The large brace introduces a **piecewise definition**: read each line as "this value, under this condition". So $|5| = 5$, $|-5| = -(-5) = 5$, $|0| = 0$. The distance between a and b on the line is $|a - b|$.

The second line of the definition, $-x$ when $x < 0$, trips people up: it looks like it makes things negative. But when x is negative, $-x$ (the opposite of x) is positive. This is the unary-minus point from Section 1.5 again.

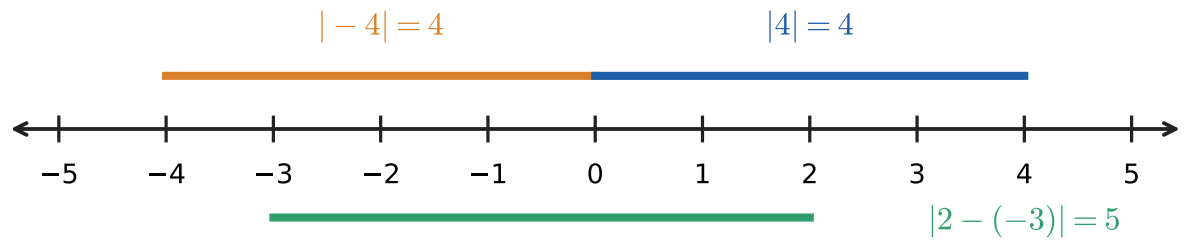


Figure 8. Absolute value as distance: from zero, or between two points.

1.7.3. Properties

For all real a, b:

$$\begin{aligned} |a| &\geq 0, \text{ and } |a| = 0 \text{ only when } a = 0 \\ |ab| &= |a|\,|b| \\ |a + b| &\leq |a| + |b| \qquad \text{(the triangle inequality)} \end{aligned}$$

The **triangle inequality** says a detour is never shorter than the direct route. Equality holds when a and b have the same sign; when they have opposite signs they partly cancel. It sounds modest but it is one of the most-used inequalities in all of analysis, and it reappears for vectors in Volume 2, Chapter 26 .

1.7.4. Worked example

The two sensors: $|(-3.2) - 1.8| = |-5| = 5$. Equivalently $|1.8 - (-3.2)| = |5| = 5$: order does not matter, which is exactly why we use absolute value for distance.

Which numbers x satisfy $|x - 2| < 3$? Read it as "the distance from x to 2 is less than 3". So x is within 3 of 2: $-1 < x < 5$. Translating absolute-value inequalities into distance statements is far more reliable than manipulating them symbolically; Chapter 5 returns to this.

1.7.5. Where you'd use this

- **Manufacturing tolerances.** A bolt is specified as 10.00 ± 0.05 mm. A bolt of diameter d passes inspection when $|d - 10.00| \le 0.05$: 9.97 mm passes (off by 0.03), 10.06 mm fails (off by 0.06). The same test checks a medicine dose, a thermostat, or a bathroom scale.
- **Measuring error.** A weather model's temperature forecasts are off by -2, $+3$, and -1 degrees. The plain average error is 0, which suggests a perfect forecast. The **mean absolute error** is $(2 + 3 + 1)/3 = 2$ degrees, which is the honest answer. Machine-learning models are trained against error measures of exactly this form (Volume 3, Chapter 31).
- **Games and robots on a grid.** A piece that moves only along rows and columns needs $|x_1 - x_2| + |y_1 - y_2|$ moves to get from one square to another: from $(2, 7)$ to $(5, 3)$ that is $3 + 4 = 7$. This **Manhattan distance** is the standard heuristic in A* path-finding on grid maps.

Computer-science connection: abs can overflow

Testing floating-point results for equality is almost always wrong (see the associativity sidebar). The standard replacement is a **tolerance**: `abs(x - y) < eps`, which is literally "the distance between x and y is less than ε". Python provides `math.isclose`, which combines an absolute and a relative tolerance.

A subtler trap: in a 32-bit signed integer type, `abs(INT_MIN)` is undefined behaviour in C and returns `INT_MIN` itself (a negative number!) in Java. The reason is that -2^{31} is representable but $+2^{31}$ is not — see Section 1.9. "The absolute value is non-negative" is a theorem about $\mathbb{Z}$, not about `int`.

US curriculum

Absolute value as distance from zero is introduced in **grade 6** (Common Core 6.NS.7). Absolute-value equations and inequalities appear in **Algebra I**, and absolute-value functions and their graphs in **Algebra I/II**.

1.8. Divisibility, primes, and factorisation

1.8.1. The problem

You need to arrange 360 chairs in a rectangular grid. What shapes are possible? You want to schedule something every 12 days and something else every 18 days; when do they coincide? A hash table works best when its size has no small factors in common with patterns in the keys. All of these are questions about how whole numbers break into products — about **divisibility**.

This section is a working toolkit. The deep theory (modular arithmetic, the Euclidean algorithm, cryptography) is Volume 3, Chapter 23 .

1.8.2. Vocabulary and notation



Division with remainder

For integers a and $b > 0$ there are unique integers q (the **quotient**) and r (the **remainder**) with

$$a = qb + r, \qquad 0 \le r < b.$$

Example: $47 = 6 \times 7 + 5$, so dividing 47 by 7 gives quotient 6, remainder 5. In code, `q, r = divmod(a, b)`, or `a // b` and `a % b`.



Divides, factor, multiple

b **divides** a, written $b \mid a$ (a vertical bar, read "b divides a"), if the remainder is 0, i.e. $a = qb$ for some integer q. Then b is a **factor** (or **divisor**) of a, and a is a **multiple** of b. So $3 \mid 12$ and $5 \nmid 12$. Do not confuse $b \mid a$ (a true/false statement) with b/a (a number).



Prime and composite

A **prime** is a natural number greater than 1 whose only positive divisors are 1 and itself: $2, 3, 5, 7, 11, 13, \ldots$. A natural number greater than 1 that is not prime is **composite**: it can be written as a product of two smaller natural numbers. 1 is neither prime nor composite. 2 is the only even prime.

1.8.3. The fundamental theorem of arithmetic

Every integer greater than 1 can be written as a product of primes, and this product is **unique** apart from the order of the factors:

$$360 = 2 \times 2 \times 2 \times 3 \times 3 \times 5 = 2^3 \cdot 3^2 \cdot 5.$$

Primes are the atoms of multiplication; the prime factorisation is a number's chemical formula. Existence is easy to see: if a number is not prime, split it into two smaller factors, and keep splitting until every piece is prime. Uniqueness is subtler and is proved in Volume 3, Chapter 23.



Why 1 is not prime

If 1 counted as prime, factorisations would not be unique: $6 = 2 \times 3 = 1 \times 2 \times 3 = 1 \times 1 \times 2 \times 3 \ldots$. Excluding 1 is not a deep fact about 1 — it is a definition chosen so that the fundamental theorem can be stated cleanly. Until the late 19th century, plenty of mathematicians did list 1 as a prime.

1.8.4. Divisibility tests

Shortcuts for spotting small factors, with the reason each works:

- **2, 5, 10:** look at the last digit. Since $10 = 2 \times 5$, every digit except the last contributes a multiple of 10, so only the units digit decides.

- **4:** last two digits form a multiple of 4 (because 100 is a multiple of 4).
- **3 and 9:** the **digit sum** is a multiple of 3 (or 9). Why: $10 = 9 + 1$, $100 = 99 + 1$, $1000 = 999 + 1$, … each power of 10 is "a multiple of 9, plus 1". So

$$\begin{aligned} 2475 &= 2 \cdot 1000 + 4 \cdot 100 + 7 \cdot 10 + 5 \\ &= 2(999 + 1) + 4(99 + 1) + 7(9 + 1) + 5 \\ &= \underbrace{(2 \cdot 999 + 4 \cdot 99 + 7 \cdot 9)}_{\text{multiple of 9}} + (2 + 4 + 7 + 5). \end{aligned}$$

  The number and its digit sum (here 18) leave the same remainder on division by 9 — and also by 3. Since 18 is a multiple of 9, so is 2475 ($= 9 \times 275$).
- **11:** alternating digit sum. Since $10 = 11 - 1$, powers of 10 alternate between "multiple of 11 plus 1" and "multiple of 11 minus 1". For 9273: $9 - 2 + 7 - 3 = 11$, a multiple of 11, so $11 \mid 9273$ ($= 11 \times 843$).

1.8.5. The method: trial division

Example 6. Procedure — prime factorisation by trial division

Input: an integer $n \geq 2$.

Steps:

1. Set the trial divisor $d = 2$.
2. While $d \times d \leq n$:
   a. While $d \mid n$: record d as a factor and replace n by n/d.
   b. Increase d by 1.
3. If $n > 1$ now, record n — it is prime.

Output: the list of prime factors of the original number, with repetition, in increasing order.

Try it: Trial division, step by step .

Two details make this correct and fast:

- **We never record a composite d.** By the time we try, say, $d = 6$, all factors 2 and 3 have already been divided out, so 6 cannot divide what remains.
- **We can stop at $\sqrt{n}$.** If $n = ab$ with both factors bigger than $\sqrt{n}$, then $ab > \sqrt{n}\sqrt{n} = n$, which is impossible. So any composite n has a factor no larger than $\sqrt{n}$. If no d up to $\sqrt{n}$ divides n, it is prime. (The test $d \times d \leq n$ is this condition without computing a square root.)

1.8.6. Worked examples

360. Each split in the tree below records one division.

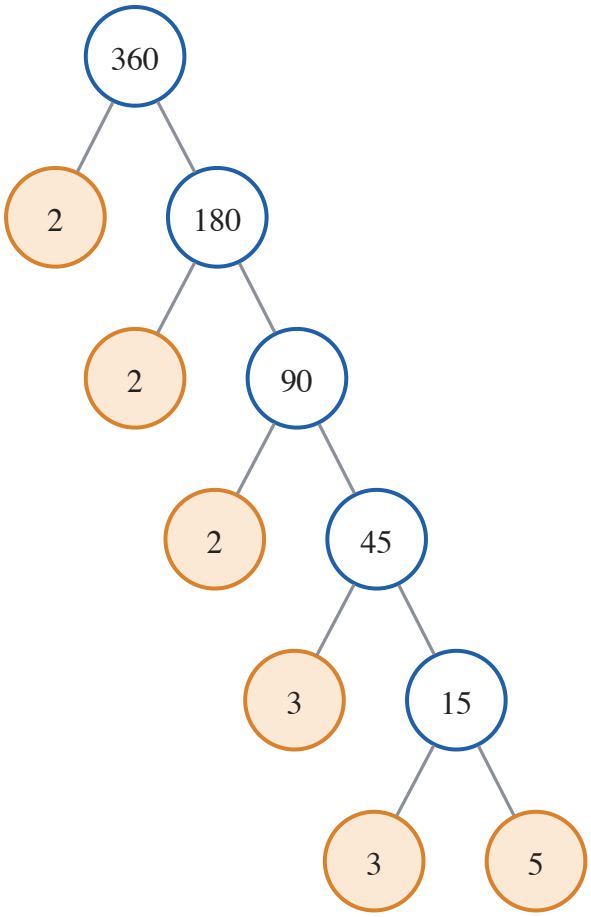


Figure 9. A factor tree for 360. The leaves (orange) are the prime factors.

By trial division: $360/2 = 180$, $180/2 = 90$, $90/2 = 45$; 2 no longer divides 45. Try 3: $45/3 = 15$, $15/3 = 5$; 3 no longer divides 5. Try 4: $4 \times 4 = 16 > 5$, stop. Remaining $5 > 1$ is prime. So $360 = 2^3 \cdot 3^2 \cdot 5$.

97. $\sqrt{97} \approx 9.85$, so only divisors 2 through 9 need checking. 97 is odd (not 2), digit sum 16 (not 3), does not end in 0 or 5 (not 5), and $97 = 13 \times 7 + 6$ (not 7). (4, 6, 8, 9 cannot divide it if 2 and 3 do not.) So 97 is prime. *Try it:* Why 97 is prime .

1001. $1001 = 7 \times 143 = 7 \times 11 \times 13$. (The alternating digit sum $1 - 0 + 0 - 1 = 0$ would have flagged 11 immediately.)

```python
def factorise(n):
    """Return the prime factors of n >= 2 in ascending order, with repetition."""
    factors = []
    d = 2
    while d * d <= n:            # only need to try divisors up to sqrt(n)
        while n % d == 0:
            factors.append(d)
            n //= d
        d += 1
    if n > 1:                    # whatever is left is itself prime
        factors.append(n)
    return factors

print(factorise(360))
print(factorise(1001))
print(factorise(97))
print(factorise(2**10 * 3 * 1_000_003))
```

```
[2, 2, 2, 3, 3, 5]
[7, 11, 13]
[97]
[2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 1000003]
```

1.8.7. Choosing a method and its limits

For numbers up to a few million, trial division is instant and is the right tool. Its cost grows like $\sqrt{n}$ steps: a 20-digit number may need 10^{10} trial divisions, and a 600-digit number is utterly beyond reach. For testing whether a large number is prime (without finding its factors) there are fast probabilistic tests (Miller–Rabin) that you will meet in Volume 3, Chapter 23 . For **factorising** large numbers, no fast method is known — and that is not a gap in this book but a gap in human knowledge on which much of internet security rests.

1.8.8. Where you'd use this

Two quantities fall straight out of factorisations. The **least common multiple** (LCM) takes each prime to the *highest* power found in either number; the **greatest common divisor** (GCD) takes each prime to the *lowest* power. (Volume 3, Chapter 23 gives a much faster way to find the GCD.)

- **Scheduling.** The opening question: one task every 12 days, another every 18. Since $12 = 2^2 \cdot 3$ and $18 = 2 \cdot 3^2$, the LCM is $2^2 \cdot 3^2 = 36$. They coincide every 36 days. The same calculation tells you when two buses with 12- and 18-minute headways leave the depot together again.
- **Tiling.** The largest square tile that exactly covers a 360 cm by 270 cm floor has side $\gcd(360, 270)$. With $360 = 2^3 \cdot 3^2 \cdot 5$ and $270 = 2 \cdot 3^3 \cdot 5$, that is $2 \cdot 3^2 \cdot 5 = 90$ cm, giving a 4×3 grid of tiles.
- **Engineering.** Two meshing gears with 12 and 18 teeth bring the same pair of teeth together every $\operatorname{lcm}(12, 18) = 36$ teeth, so any flaw on one tooth keeps hitting the same few teeth on the other. Designers prefer counts with no common factor (a "hunting tooth"): 13 and 18 teeth meet again only after $13 \times 18 = 234$ teeth, which spreads the wear.

- **Error detection.** An ISBN-10 is valid when $10d_1 + 9d_2 + \cdots + 1 \cdot d_{10}$ is divisible by 11. For 0-306-40615-2 the sum is 132 = 12×11, which is valid. Mistype any single digit, or swap two adjacent digits, and the sum stops being a multiple of 11.

Computer-science connection: primes in the wild

- **Cryptography.** RSA keys are built by multiplying two huge random primes. Multiplying is easy; recovering the primes from the product is (as far as anyone knows) infeasible for classical computers.
- **Hashing.** Hash-table sizes and multiplicative hash constants are often prime, so that regular patterns in keys (all multiples of 8, say) do not pile up in the same buckets.
- **Negative remainders.** Languages disagree on division with remainder for negative numbers. Python: `-7 // 2 == -4`, `-7 % 2 == 1` (remainder always has the sign of the divisor, matching the mathematical convention $0 \le r < b$). C, Java, JavaScript: `-7 / 2 == -3`, `-7 % 2 == -1`. Code like `if (n % 2 == 1)` to test oddness silently fails for negative `n` in C.



Factor explorer

Enter a number and watch trial division run step by step: the current divisor, the shrinking value, the growing factor list, and a marker showing $\sqrt{n}$ where the search may stop. Then plot the number of steps against n for n up to a million and see primes (worst case) form the top envelope.

Try it: Open the factor explorer · Factorise any number

1.8.9. Exercises

1. Is 2,345,678 divisible by 2? by 3? by 9? by 11? Justify with the tests.
2. Find the prime factorisation of 1260 and of 2310.
3. Why does trial division never need to try $d = 15$ separately, in the sense that 15 can never divide the remaining n when it is reached?
4. How many different rectangular grids (rows × columns, order mattering) can 36 chairs form? Use the factorisation $36 = 2^2 \cdot 3^2$ to count without listing.

▼ **Answers**

1. Last digit 8: divisible by 2. Digit sum $2+3+4+5+6+7+8 = 35$: not a multiple of 3, so neither 3 nor 9. Alternating sum from the right: $8-7+6-5+4-3+2 = 5$: not a multiple of 11.
2. $1260 = 2 \cdot 630 = 2^2 \cdot 315 = 2^2 \cdot 3 \cdot 105 = 2^2 \cdot 3^2 \cdot 35 = 2^2 \cdot 3^2 \cdot 5 \cdot 7$.
   $2310 = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11$.

3. By the time $d = 15$, every factor 3 and every factor 5 has been divided out of n (at $d = 3$ and $d = 5$). A number with no factor 3 cannot be a multiple of 15.
4. A divisor of 36 has the form $2^i 3^j$ with $i \in \{0, 1, 2\}$ and $j \in \{0, 1, 2\}$: $3 \times 3 = 9$ divisors. Each divisor r gives the grid "r rows × $36/r$ columns", so there are 9 grids (1×36, 2×18, 3×12, 4×9, 6×6, 9×4, 12×3, 18×2, 36×1). Counting by exponents like this is a first taste of Volume 3, Chapter 24 .

US curriculum

Factors, multiples, and primes appear in **grade 4** (Common Core 4.OA.4); prime factorisation, greatest common factor, and least common multiple in **grade 6** (6.NS.4). The fundamental theorem of arithmetic is usually stated without proof; the proof belongs to college **discrete mathematics** or **number theory**.

1.9. Integers inside a computer: two's complement and overflow

1.9.1. The problem

In mathematics, $\mathbb{Z}$ is infinite: you can always add 1. In a computer, an `int32` has exactly 2^{32} possible bit patterns. So what happens when you add 1 to the largest one? And how does a register that only holds bits represent -5 at all?

In December 2014 the YouTube view counter for the video "Gangnam Style" approached $2\,147\,483\,647 = 2^{31} - 1$, the largest value of a signed 32-bit integer. YouTube upgraded the counter to 64 bits before it overflowed. Many systems have not been so lucky; the "Year 2038 problem" is the same story for 32-bit timestamps counting seconds since 1 January 1970.

1.9.2. Intuition: a clock, not a line

A fixed-width integer does not live on a number line; it lives on a **circle**, like the hours on a clock. With 4 bits there are 16 patterns, 0000 to 1111. Adding 1 moves one step clockwise. After 1111 comes 0000 again — the carry out of the top bit is simply lost.

The question is just how to *label* the 16 positions. Unsigned integers label them 0 to 15. **Two's complement** keeps 0 to 7 for the patterns with top bit 0 and labels the other half -8 to -1, so that stepping counter-clockwise from 0 gives -1, -2, … — exactly as on the number line.

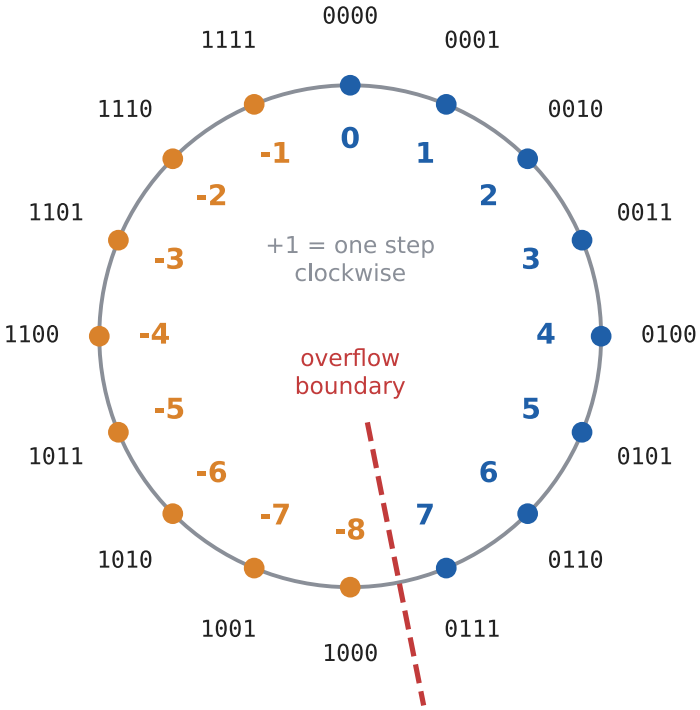


Figure 10. 4-bit two's complement. Each pattern is a position on a circle; adding 1 steps clockwise. The jump from 7 to -8 is overflow.

1.9.3. Why it works

On the 16-position clock, moving 16 steps brings you back to where you started, so the patterns cannot tell the difference between x and $x + 16$ or $x - 16$. Mathematicians say the arithmetic is **modulo 16** (Volume 3, Chapter 23 develops this as modular arithmetic). The pattern for -3 is therefore the same as the pattern for $16 - 3 = 13 = 1101_2$. *Try it:* Step past 7 on the clock ; –3 is 1101 in 4 bits .

In general, with n bits, the negative number $-x$ is stored as the bit pattern of $2^n - x$. And since $2^n - x = (2^n - 1 - x) + 1$, and $2^n - 1$ is all ones ($1111_2 = 15$), subtracting x from all-ones simply flips every bit. Hence the famous rule:

Example 7. Procedure — negate a two's-complement integer

Input: the n-bit pattern of an integer x.

Steps:

1. Invert every bit (0 ↔ 1). This computes $2^n - 1 - x$.
2. Add 1, discarding any carry out of the top bit. This gives $2^n - x$.

Output: the n-bit pattern of $-x$.

Worked example (8 bits). Represent -5: *Try it:* –5 in 8 bits .

$$5 = 0000\,0101_2$$
$$\text{invert} \to 1111\,1010_2$$
$$\text{add } 1 \to 1111\,1011_2 = \mathrm{FB}_{16}.$$

Check: unsigned, $\mathrm{FB}_{16} = 251 = 256 - 5$. ✓ And $5 + (-5)$: $0000\,0101 + 1111\,1011 = 1\,0000\,0000$; the ninth bit falls off, leaving $0000\,0000$. ✓

The beauty of the scheme is that the hardware adder does not need to know whether numbers are signed: the same circuit adds unsigned and two's-complement integers correctly, because both are just arithmetic modulo 2^n with different labels.

1.9.4. Vocabulary



Two's complement, overflow, signed and unsigned

An **unsigned** n-bit integer represents 0 to $2^n - 1$. A **signed** n-bit integer in **two's complement** represents -2^{n-1} to $2^{n-1} - 1$; the top bit is the **sign bit** (1 = negative). **Overflow** is when the true mathematical result of an operation lies outside the representable range, so the stored result wraps around. For 32 bits the signed range is $-2\,147\,483\,648$ to $2\,147\,483\,647$.

The range is lopsided — one more negative than positive — because zero takes one of the "non-negative" patterns. This is why $-(-2^{31})$ overflows back to -2^{31}, and why `abs(INT_MIN)` misbehaves.

1.9.5. Simulating it

Python's integers never overflow, which makes them a good laboratory: we can simulate a 32-bit register by keeping only the low 32 bits.

```python
def wrap_int32(x):
    """Reduce x to what a 32-bit two's-complement register would hold."""
    x &= 0xFFFFFFFF                  # keep the low 32 bits
    return x - (1 << 32) if x & 0x80000000 else x   # top bit set => negative

big = 2_147_483_647                  # 2^31 - 1, the largest int32
print(wrap_int32(big + 1))
print(wrap_int32(-big - 1 - 1))
print(wrap_int32(-5), hex(-5 & 0xFFFFFFFF))
```

Try it: Watch int32 wrap around .

```
-2147483648
2147483647
-5 0xfffffffb
```

1.9.6. Failure modes



Overflow is silent

Most languages do not stop when integers overflow. In C and C++ signed overflow is **undefined behaviour** — the compiler may assume it never happens and optimise accordingly, sometimes deleting your overflow check. Java and Go wrap silently. Rust panics in debug builds and wraps in release builds. A classic bug: the midpoint `(lo + hi) / 2` in binary search overflows when `lo + hi > INT_MAX`; the safe form is `lo + (hi - lo) / 2`. That bug sat in the Java standard library's binary search for about nine years before it was found in 2006.

Computer-science connection: choosing integer widths

Rules of thumb that follow from this section:

- Counts that can only grow (views, bytes transferred, IDs) should be 64-bit from day one. $2^{63} - 1 \approx 9.2 \times 10^{18}$ is effectively unreachable; $2^{31} - 1 \approx 2.1 \times 10^{9}$ is reached by successful products.
- Use unsigned types for bit manipulation and sizes; be careful mixing them with signed values, since in C `-1 < 0u` is false (the -1 is converted to $2^{32} - 1$).
- Wraparound is a **feature** for hashing, checksums, and random-number generators, which are deliberately computed modulo 2^{32} or 2^{64}.

Try it: Why −1 < 0u is false.



Ones' complement and the two zeros

Early computers tried other schemes. **Sign-magnitude** used the top bit as a minus sign; **ones' complement** negated by flipping bits without adding 1. Both have two zeros ($+0$ and -0) and need special adder logic. Two's complement won because the plain unsigned adder just works. Floating point, interestingly, still uses sign-magnitude, which is why IEEE 754 has both `0.0` and `-0.0`.



Overflow clock

An n-bit clock with a slider for n from 3 to 8. Each position shows its bit pattern and both its unsigned and signed labels. Add or subtract a chosen amount and watch the hand move; the overflow boundary flashes when crossed. A second mode multiplies, showing that multiplication modulo 2^n also wraps.

Try it: Multiplication wraps too · Open the Overflow clock

1.9.7. Exercises

1. What range of values does a signed 8-bit integer hold? An unsigned 16-bit integer?

2. Find the 8-bit two's-complement pattern for -1, -128, and -100.
3. What signed 8-bit value does the pattern $1001\,0110_2$ represent?
4. In 8-bit signed arithmetic, compute $100 + 50$. What value is stored, and why?

▼ **Answers**

1. Signed 8-bit: $-2^7 = -128$ to $2^7 - 1 = 127$. Unsigned 16-bit: 0 to $2^{16} - 1 = 65\,535$.
2. -1: invert $0000\,0001$ to $1111\,1110$, add 1: $1111\,1111$. -128: $256 - 128 = 128 = 1000\,0000$. -100: $256 - 100 = 156 = 1001\,1100$.
3. Top bit is 1, so negative. Unsigned value $128 + 16 + 4 + 2 = 150$; signed value $150 - 256 = -106$. (Or: invert to $0110\,1001 = 105$, add 1: 106, so the value is -106.)
4. The true sum 150 exceeds 127. The stored pattern is that of 150, which read as signed is $150 - 256 = -106$. The result wrapped around the clock.

US curriculum

Two's complement is not part of the standard US mathematics sequence. It is taught in **AP Computer Science** (lightly) and in college **computer organisation / architecture** courses. The underlying modular arithmetic appears in **discrete mathematics**.

1.10. Estimation and rounding

1.10.1. The problem

A cloud bill says a job processed 3.8 TB at $0.023 per GB. Is the $87.40 charge plausible? A colleague claims a loop over 50,000 records with a 2 ms database call each "takes a couple of seconds". Right? You do not need a calculator to answer either; you need **estimation**. Estimation is how you catch errors — your own, a spreadsheet's, a program's — before they cost you.

1.10.2. Rounding



Rounding, significant figures, truncation

To **round** a number is to replace it by a nearby number with fewer digits. Rounding "to the nearest 10", "to 2 decimal places" (2 d.p.), or "to 3 significant figures" (3 s.f.) specifies which digit is the last one kept. The standard rule: look at the first digit dropped; if it is 5 or more round up, otherwise round down. **Truncation** simply drops digits (always toward zero). The symbol $\approx$ means "approximately equal to".

Example 8. Procedure — round to a given place

> **Input:** a number x and the place to keep (e.g. hundredths).
>
> **Steps:**
>
> 1. Identify the **rounding digit** — the last digit to keep.
> 2. Look at the digit immediately to its right (the **decider**).
> 3. If the decider is 0–4, keep the rounding digit. If 5–9, increase it by 1 (carrying if it was 9).
> 4. Drop all digits after the rounding digit (replacing them with zeros if they are to the left of the decimal point).
>
> **Output:** the rounded number.

Worked examples. 3.14159 to 2 d.p.: rounding digit 4, decider 1, so 3.14. 2.7183 to 1 d.p.: rounding digit 7, decider 1, so 2.7. $47\,862$ to the nearest thousand: rounding digit 7, decider 8, so $48\,000$. 9.96 to 1 d.p.: rounding digit 9, decider 6, so the 9 becomes 10, carry: 10.0 (keep the trailing zero to show the precision).

Significant figures count from the first non-zero digit: $0.004\,072$ to 2 s.f. is 0.0041. They get a fuller treatment alongside scientific notation in Chapter 3.

1.10.3. Alternative rounding rules

"Round half up" is biased: over many values ending in exactly 5, it always goes up, nudging totals upward. **Round half to even** ("banker's rounding") sends an exact half to whichever neighbour is even: $2.5 \to 2$, $3.5 \to 4$. Over many values, the ups and downs cancel. IEEE 754 floating-point uses round-half-to-even by default, and so does Python's `round`:

```
>>> round(2.5), round(3.5), round(0.125, 2), round(2.675, 2)
(2, 4, 0.12, 2.67)
```

PYTHON

The last one is not banker's rounding: 2.675 cannot be stored exactly in binary and is actually stored as a value slightly *below* 2.675, so it rounds down. That is a story for Chapter 2.

1.10.4. Choosing a method

- Everyday reporting: round half up — it is what readers expect.
- Accumulating many rounded values (financial ledgers, statistics): round half to even, or better, do not round until the end.
- Converting to an integer index or a count of whole items: truncation or floor, deliberately.



Rounding too early

Round only at the end. Rounding intermediate results compounds errors: $0.4 + 0.4 + 0.4 = 1.2$, which rounds to 1, but rounding each term first gives $0 + 0 + 0 = 0$. **Double rounding** is a related trap: 1.45 to 1 d.p. is 1.5 (half up), then to the nearest whole is 2; but 1.45 rounded directly to the nearest whole is 1.

1.10.5. Estimation

Estimation means replacing numbers by round values that are easy to work with, computing, and judging whether the true answer is close.

The cloud bill. $3.8\ \text{TB} \approx 4000\ \text{GB}$ (rounding up a little). $\$0.023 \approx \$0.02$ (rounding down a little — the errors partly cancel). $4000 \times 0.02 = 80$. So about \$80–\$90: the \$87.40 charge is plausible. (Exactly: $3800 \times 0.023 = 87.40$.)

The loop. $50\,000 \times 2\ \text{ms} = 100\,000\ \text{ms} = 100\ \text{s}$, nearly two minutes, not "a couple of seconds". The colleague is out by a factor of about 50 — the kind of error estimation is built to catch.

Order-of-magnitude (Fermi) estimates. How many piano tuners are in Chicago? Enrico Fermi famously posed such questions to students: chain together rough guesses (people, households, pianos per household, tunings per year, tunings per tuner per year), each rounded to a power of ten. Individual errors tend to partially cancel, and the result is often within a factor of 3 of the truth. Chapter 3 develops the power-of-ten thinking that makes this work.

Computer-science connection: back-of-the-envelope engineering

System design is estimation. "Latency numbers every programmer should know" (an L1 cache hit is about a nanosecond, a main-memory reference about 100 ns, a same-datacentre round trip about half a millisecond, a disk seek about 10 ms) let you estimate before you build: will this fit in memory, is the bottleneck network or disk, can one machine handle the load? An estimate that is right to within a factor of 2 will usually tell you which design to pursue.



Estimation trainer

The app shows a quick calculation (387×52, 0.023×3800, $7.9\ \text{GB}/41\ \text{s}$) for three seconds, then asks for an estimate. Score by how close the estimate is on a logarithmic scale, and track your typical error factor over time.

1.10.6. Exercises

1. Round 6.0451 to (a) 2 d.p., (b) 1 d.p., (c) 3 s.f., (d) the nearest whole number.
2. Estimate 48.7×21.3, then compute it exactly. What was your relative error?
3. A service handles 1,200 requests per second, each producing a 3 KB log line. Roughly how much log data is produced per day?

▼ **Answers**

1. (a) 6.05 (decider 5, round up). (b) 6.0 (decider 4). (c) 6.05. (d) 6.
2. $50 \times 20 = 1000$. Exactly: $48.7 \times 21.3 = 1037.31$. Relative error $(1037.31 - 1000)/1037.31 \approx 0.036$, about 3.6%. (Rounding 48.7 up and 21.3 down partly cancelled.)
3. $1200 \times 3\ \text{KB} = 3600\ \text{KB/s} \approx 3.6\ \text{MB/s}$. A day is $86\,400 \approx 10^5$ seconds. $3.6\ \text{MB} \times 10^5 \approx 3.6 \times 10^5\ \text{MB} \approx 360\ \text{GB}$ per day. (Exactly, with 86,400 s: about 311 GB.)

US curriculum

Rounding to a place value is a **grade 3–5** topic (Common Core 3.NBT.1, 5.NBT.4). Estimation recurs throughout middle school. Significant figures are usually taught in high-school **chemistry** and **physics** rather than in mathematics courses.

1.11. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Number theory
- Where floating point bites
- Floating-point inspector
- Grammar and parser playground
- Number theory step by step
- Programmer's calculator

1.12. Mastery check

1. Classify each number as natural, integer, rational, or irrational (give the smallest set it belongs to): -4, 0.75, $\sqrt{9}$, $\sqrt{10}$, $0.\overline{6}$.
2. Convert $1101\,0110_2$ to hex and to decimal. Convert 300 to binary.
3. Compute 49×51 mentally and name every arithmetic law you used.
4. Prove that $(-a)(-b) = ab$ starting from $(-1)(-1) = 1$ and the laws of arithmetic.
5. Evaluate $12 - 2 \times (1 - 4)^2 \div 6$. Then state what Python prints for `-4 ** 0.5 + 10 // -3`.
6. Find the prime factorisation of 2024 and explain why trial division can stop where it does.
7. What is the 8-bit two's-complement pattern of -37? What happens in 8-bit signed arithmetic if you compute $127 + 1$?
8. Estimate the number of seconds in a year to one significant figure, then to three.

Decision rule: All correct without hesitation → move on to Chapter 2. Stuck on 1–2 → reread the matching section (Section 1.1, Section 1.3, Section 1.4, Section 1.5, Section 1.6, Section 1.8, Section 1.9, Section 1.10). Stuck on more → work the section exercises first.

▼ **Mastery check answers**

1. -4: integer. $0.75 = \frac{3}{4}$: rational. $\sqrt{9} = 3$: natural. $\sqrt{10}$: irrational (10 is not a perfect square; the argument for $\sqrt{2}$ adapts). $0.\overline{6} = \frac{2}{3}$: rational.
2. Group bits: $1101 = \mathrm{D}$, $0110 = 6$, so $\mathrm{D6}_{16}$. Decimal: $13 \times 16 + 6 = 214$. For 300: 256 fits (44 left), 128 no, 64 no, 32 fits (12 left), 16 no, 8 fits (4 left), 4 fits (0 left), 2 no, 1 no: $1\,0010\,1100_2$. Check: $256 + 32 + 8 + 4 = 300$. ✓
3. $49 \times 51 = 49 \times (50 + 1) = 49 \times 50 + 49$ (distributive) $= 2450 + 49 = 2499$. Alternatively $(50 - 1)(50 + 1) = 2500 - 1 = 2499$ (distributive twice, commutative, and inverse to cancel $-50 + 50$).
4. $(-a)(-b) = ((-1)a)((-1)b)$ (since $-x = (-1)x$) $= (-1)(-1)\,ab$ (commutative and associative laws to regroup) $= 1 \cdot ab = ab$ (identity).
5. Brackets: $1 - 4 = -3$; exponent: $(-3)^2 = 9$; then $2 \times 9 = 18$, $18 \div 6 = 3$; finally $12 - 3 = 9$. Python: `-4 ** 0.5` is $-(4^{0.5}) = -2.0$; `10 // -3` floors $-3.33\ldots$ to -4; the sum is `-6.0`.
6. $2024 = 2 \cdot 1012 = 2^2 \cdot 506 = 2^3 \cdot 253$; 253 is not divisible by 3, 5, or 7; $253 = 11 \times 23$. So $2024 = 2^3 \cdot 11 \cdot 23$. After dividing out 11 the remainder is 23, and the next trial divisor 12 has $12^2 = 144 > 23$, so 23 must be prime: a composite number always has a factor no larger than its square root.
7. $37 = 0010\,0101$; invert: $1101\,1010$; add 1: $1101\,1011$ ($= \mathrm{DB}_{16} = 219 = 256 - 37$). $127 + 1$ overflows: the pattern $1000\,0000$ is read as -128.

8. $365 \times 24 \times 3600$. Roughly $400 \times 25 \times 4000 = 4 \times 10^7$ is a little high (all three rounded up); more carefully $365 \times 86\,400 = 31\,536\,000 \approx 3 \times 10^7$ (1 s.f.) or 3.15×10^7 (3 s.f.). A handy fact: a year is about $\pi \times 10^7$ seconds.

Chapter 2. Fractions, Decimals, Percentages, and Ratios

Chapter map: fractions are where arithmetic first stops being automatic, which is why so many adults quietly avoid them. We fix that by giving each rule a reason: why you need a common denominator to add, why "flip and multiply" works, why some fractions become terminating decimals and others repeat forever, and why $0.999\ldots$ *is exactly 1. Then we turn to the fraction's everyday disguises — percentages, ratios, rates, and unit conversions — and finish with the reason your computer insists that* $0.1 + 0.2 \neq 0.3$.

Why it matters: scaling a recipe, reading a nutrition label, judging a "30% off" sale, working out a drug dose, interpreting an A/B test, and storing money in a database are all fraction problems in disguise. People who are shaky on fractions lose money and make bad decisions on all of them, and programmers who are shaky on them ship rounding bugs.

Where this chapter sits

Prerequisites: Chapter 1 — especially the arithmetic laws (Section 1.4), negative numbers, and prime factorisation (Section 1.8).

This chapter unlocks: every later chapter. Algebraic fractions (Chapter 4, Chapter 7) obey exactly the rules developed here with letters in place of numbers. Rates of change are the seed of the derivative (Volume 2, Chapter 17). Probabilities are fractions (Volume 3, Chapter 29). Floating-point representation returns in depth in Volume 2, Chapter 32 .

Fast path: if you can do the mastery check at the end quickly and correctly, skip to Chapter 3. Even then, read Section 2.8; it is the part most programmers half-know.

2.1. What a fraction is

2.1.1. The problem

"Three quarters" sounds simple, but the symbol $\frac{3}{4}$ is asked to do at least three different jobs, and much of the confusion people have with fractions comes from switching between them without noticing.

2.1.2. Intuition: three readings of one symbol

1. **Part of a whole.** Cut something into 4 equal parts and take 3.
2. **Division.** Share 3 identical things equally among 4 people; each gets $\frac{3}{4}$ of a thing. So $\frac{3}{4} = 3 \div 4$.
3. **A point on the number line.** Split the segment from 0 to 1 into 4 equal steps and walk 3 of them. $\frac{3}{4}$ is a single number with a definite position, just like 2 or -5.

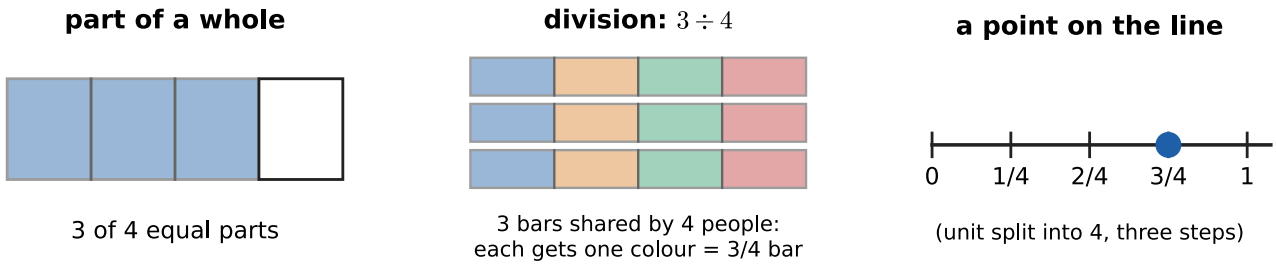


Figure 11. Three readings of $\frac{3}{4}$: a part of a whole, the answer to a division, and a location on the number line.

The second reading is the one that makes the rest of the chapter work. Once you accept that $\frac{a}{b}$ *means* $a \div b$ — "the number which, multiplied by b, gives a" — every fraction rule can be derived from the arithmetic laws of Section 1.4. The third reading is the one that matters most later: fractions are not a separate kind of object, they are numbers, and they live on the same line as everything else.

2.1.3. Vocabulary and notation

Numerator, denominator, fraction bar

In $\frac{a}{b}$ the top number a is the **numerator** (it *numbers*, or counts, the parts) and the bottom number b is the **denominator** (it *denominates*, or names, the size of the parts: quarters, eighths). The line between them is the **fraction bar** or **vinculum**, and it means division. a/b (with a slash, or **solidus**) means the same thing. The denominator can never be 0 (Section 1.4).

Proper, improper, unit fraction, mixed number

A **proper fraction** has numerator smaller than denominator in size ($\frac{3}{4}$); its value lies between -1 and 1. An **improper fraction** has numerator at least as large as denominator ($\frac{7}{4}$). A **unit fraction** has numerator 1 ($\frac{1}{4}$). A **mixed number** writes an improper fraction as a whole number plus a proper fraction: $1\frac{3}{4} = 1 + \frac{3}{4} = \frac{7}{4}$. "Improper" is not an insult; in algebra improper fractions are usually *preferred*.

Reciprocal

The **reciprocal** of a non-zero number x is $\frac{1}{x}$, the number you multiply x by to get 1. The reciprocal of $\frac{a}{b}$ is $\frac{b}{a}$ ("turn it upside down"), because $\frac{a}{b} \cdot \frac{b}{a} = \frac{ab}{ba} = 1$. Zero has no reciprocal.

Where the bar came from

The horizontal fraction bar was used by Arab mathematicians (notably al-Hassār, 12th century) and popularised in Europe by Fibonacci's *Liber Abaci* (1202). The slash form a/b arrived much later, largely because typesetters wanted fractions to fit on a single line of type — the same reason programming languages use it. Ancient Egyptians wrote almost every fraction as a sum of distinct *unit* fractions: $\frac{3}{4} = \frac{1}{2} + \frac{1}{4}$.

US curriculum

Fractions as parts of a whole and as points on the number line: **grade 3** (Common Core 3.NF). Fractions as division: **grade 5** (5.NF.3). This is one of the most heavily emphasised strands in US elementary standards, precisely because difficulty with fractions predicts difficulty in **Algebra I**.

2.2. Equivalent fractions and simplifying

2.2.1. The problem

$\frac{1}{2}$, $\frac{2}{4}$, $\frac{4}{8}$, and $\frac{50}{100}$ are written differently but name the same point on the number line. Which version should we use, and how do we get from one to another?

2.2.2. Intuition

Cutting every piece in half doubles the number of pieces *and* doubles the number you have taken. The amount does not change.

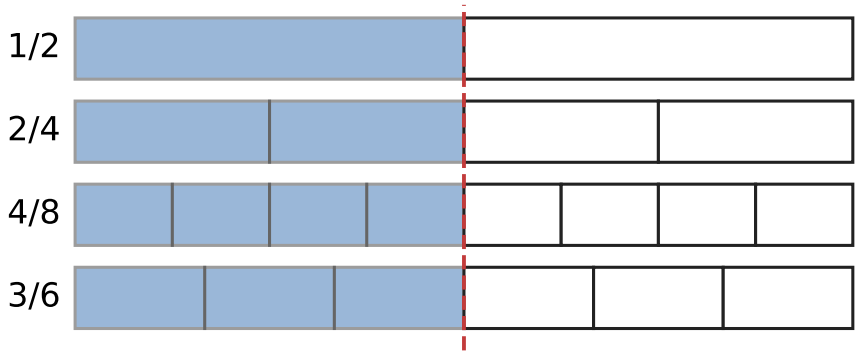


Figure 12. Equivalent fractions: cutting each part into smaller equal parts changes the numbers but not the amount.

2.2.3. Why it works

Multiplying top and bottom by the same non-zero number k is the same as multiplying by $\frac{k}{k} = 1$, which changes nothing (1 is the multiplicative identity):

$$\frac{a}{b} = \frac{a}{b} \cdot 1 = \frac{a}{b} \cdot \frac{k}{k} = \frac{ak}{bk}.$$

(We use the multiplication rule $\frac{a}{b} \cdot \frac{c}{d} = \frac{ac}{bd}$ here; it is justified in Section 2.3.) Read backwards, the same equation says we may **cancel** a common factor of top and bottom. A fraction is **in lowest terms** (or **simplified**, or **reduced**) when numerator and denominator have no common factor other than 1.

2.2.4. Vocabulary



Greatest common divisor and least common multiple

The **greatest common divisor** $\gcd(a, b)$ (also called greatest common factor, GCF, or highest common factor, HCF) is the largest positive integer dividing both a and b. The **least common multiple** $\operatorname{lcm}(a, b)$ is the smallest positive integer that both divide. Example: $\gcd(12, 18) = 6$, $\operatorname{lcm}(12, 18) = 36$. For positive integers, $\gcd(a, b) \cdot \operatorname{lcm}(a, b) = ab$. Two numbers with gcd 1 are **coprime** (or **relatively prime**).

2.2.5. The method

Example 9. Procedure — simplify a fraction

Input: a fraction $\frac{p}{q}$ with integers $p, q \neq 0$.

Steps:

1. Compute $g = \gcd(p, q)$.
2. Divide both numerator and denominator by g.
3. If the denominator is negative, multiply top and bottom by -1 so the sign sits on the numerator.

Output: the equivalent fraction in lowest terms.

There are two good ways to find the gcd.

By prime factorisation (from Section 1.8): take each prime to the *smaller* of its two exponents. (For the lcm, take the *larger*.)

By Euclid's algorithm: $\gcd(a, b) = \gcd(b, r)$ where r is the remainder of a divided by b; repeat until the remainder is 0, and the last non-zero remainder is the gcd. It works because any number dividing both a and b also divides $r = a - qb$, and vice versa, so the pair (b, r) has exactly the same common divisors as (a, b) — but smaller numbers. Volume 3, Chapter 23 treats it fully.

2.2.6. Worked example

Simplify $\frac{84}{126}$. *Try it:* 84/126 by Euclid .

Method 1, factorisation. $84 = 2^2 \cdot 3 \cdot 7$ and $126 = 2 \cdot 3^2 \cdot 7$. Smaller exponents: $2^1 \cdot 3^1 \cdot 7^1 = 42$. So

$$\frac{84}{126} = \frac{2 \cdot 42}{3 \cdot 42} = \frac{2}{3}.$$

Method 2, Euclid.

$$126 = 1 \times 84 + 42 \qquad \gcd(126, 84) = \gcd(84, 42)$$
$$84 = 2 \times 42 + 0 \qquad \text{remainder 0: the gcd is 42.}$$

Method 3, cancel in stages. Both even: $\frac{42}{63}$. Both divisible by 3 (digit sums 6 and 9): $\frac{14}{21}$. Both divisible by 7: $\frac{2}{3}$. No common factor remains.

2.2.7. Choosing a method

- Small numbers whose factors you can see: **cancel in stages**. Fast, and each step is easy to check.
- Numbers whose factorisations you already know (or will need anyway, e.g. for an lcm): **factorisation**.
- Large numbers, or in code: **Euclid**. It needs no factorisation at all, which matters because factorising is hard (Section 1.8) while Euclid takes only a handful of steps even for huge numbers.



Cancelling terms instead of factors

You may cancel a common **factor** (something multiplied), never a common **term** (something added). $\frac{2 \cdot 5}{2 \cdot 7} = \frac{5}{7}$ is fine. $\frac{2+5}{2+7} = \frac{7}{9}$, not $\frac{5}{7}$. In algebra this becomes the classic error $\frac{x+3}{x+5} \to \frac{3}{5}$, which is wrong for almost every x. If in doubt, test with numbers.

Computer-science connection: canonical forms

Simplifying is an example of reducing a value to a **canonical form**: one standard representative of each equivalence class. Python's `fractions.Fraction(84, 126)` stores `2/3`, so equality testing and hashing become simple comparisons of the stored pair. The same idea appears in Unicode normalisation, URL canonicalisation, sorted-tuple keys for sets, and normal forms in databases and logic. Whenever many representations mean the same thing, pick one and convert to it on entry.

2.3. Arithmetic with fractions

2.3.1. The problem

$\frac{1}{3} + \frac{1}{4} = ?$ The tempting answer $\frac{2}{7}$ is wrong: $\frac{2}{7}$ is *smaller* than $\frac{1}{3}$, and adding a positive amount cannot make something smaller. Meanwhile multiplication, which feels harder, has the "naive" rule: $\frac{1}{3} \times \frac{1}{4} = \frac{1}{12}$. Why the asymmetry?

2.3.2. Adding and subtracting: why a common denominator?

You can only add counts of the **same kind of thing**. Three apples plus two apples is five apples; three apples plus two oranges is not five of anything until you re-describe both as "fruit". A third and a quarter are pieces of different sizes. To add them we re-cut both into pieces of a common size — twelfths, since both 3 and 4 divide 12.



Figure 13. Adding $\frac{1}{3}+\frac{1}{4}$: re-cut into twelfths so the pieces are the same size, then count them.

$$\frac{1}{3}+\frac{1}{4}=\frac{4}{12}+\frac{3}{12}=\frac{7}{12}.$$

Once the denominators match, adding is just the distributive law: $\frac{a}{d}+\frac{b}{d}=a\cdot\frac{1}{d}+b\cdot\frac{1}{d}=(a+b)\cdot\frac{1}{d}=\frac{a+b}{d}$. That is the whole reason: you can factor out a common piece size, but only if it is common.

Example 10. Procedure — add or subtract fractions

Input: fractions $\frac{a}{b}$ and $\frac{c}{d}$.

Steps:

1. Choose a common denominator L: the least common multiple $\operatorname{lcm}(b,d)$ (smallest numbers), or simply bd (always works).
2. Rewrite each fraction with denominator L: multiply top and bottom of $\frac{a}{b}$ by L/b, and of $\frac{c}{d}$ by L/d.
3. Add (or subtract) the numerators; keep the denominator L.
4. Simplify.

Output: the sum (or difference) in lowest terms.

Using $L=bd$ gives a formula worth knowing by heart, because it is the one used with letters in algebra:

$$\frac{a}{b}+\frac{c}{d}=\frac{ad}{bd}+\frac{bc}{bd}=\frac{ad+bc}{bd}.$$

Worked example. $\frac{5}{12}+\frac{7}{18}$.

$$\begin{aligned}
\operatorname{lcm}(12,18) &= 36 && (12 = 2^2\cdot 3,\ 18 = 2\cdot 3^2,\ \text{larger exponents } 2^2\cdot 3^2)\\
\frac{5}{12}+\frac{7}{18} &= \frac{5\cdot 3}{12\cdot 3}+\frac{7\cdot 2}{18\cdot 2} && (36/12 = 3,\ 36/18 = 2)\\
&= \frac{15}{36}+\frac{14}{36}\\
&= \frac{29}{36}. && (29 \text{ is prime: already in lowest terms})
\end{aligned}$$

With $L = bd$ instead: $\frac{5\cdot 18+12\cdot 7}{12\cdot 18} = \frac{90+84}{216} = \frac{174}{216}$, which simplifies (divide by 6) to $\frac{29}{36}$. Same answer; more simplifying.

Choosing: the lcm keeps numbers small and is best by hand when the denominators share factors. The product bd needs no thought and is best when the denominators are coprime (where it *is* the lcm) and in algebra, where you usually cannot find an lcm of letters anyway.

2.3.3. Multiplying: "of"

$\frac{2}{3}\times\frac{3}{4}$ means "two thirds of three quarters". Take a unit square. Shade $\frac{3}{4}$ of its width and $\frac{2}{3}$ of its height: the overlap is a 2×3 block of small rectangles, out of a $3\times 4 = 12$ grid. So

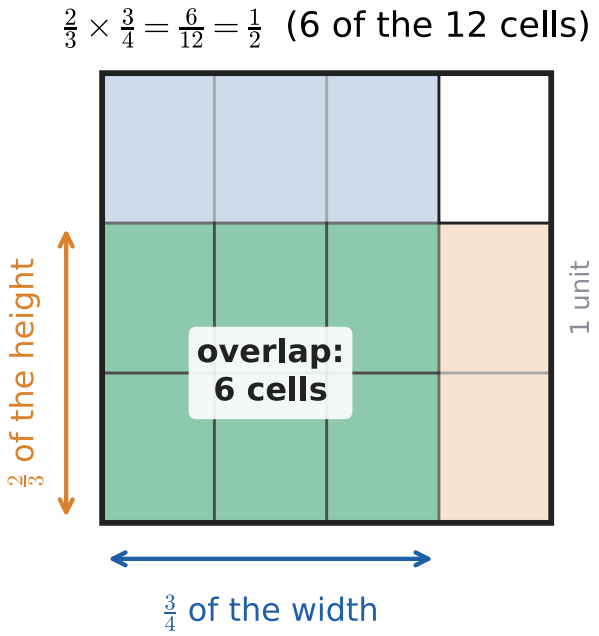


Figure 14. $\frac{2}{3}\times\frac{3}{4}$ as area: cutting the unit square into $3\times 4 = 12$ cells, the overlap of three quarters of the width with two thirds of the height is $2\times 3 = 6$ cells.

$$\frac{a}{b}\times\frac{c}{d} = \frac{ac}{bd}, \qquad \frac{2}{3}\times\frac{3}{4} = \frac{6}{12} = \frac{1}{2}.$$

Multiplication does not need a common denominator because it is not counting same-sized pieces; it is scaling one quantity by another, and the grid of small rectangles automatically has bd cells. A practical tip: **cancel before multiplying**. $\frac{2}{3}\times\frac{3}{4}$: the 3s cancel, and the 2 cancels into the 4, giving $\frac{1}{2}$ without ever forming 12.

2.3.4. Dividing: why "flip and multiply" works

$\frac{3}{2} \div \frac{1}{4}$ asks: "how many quarters fit into one and a half?" Each whole holds four quarters, so one and a half holds six.

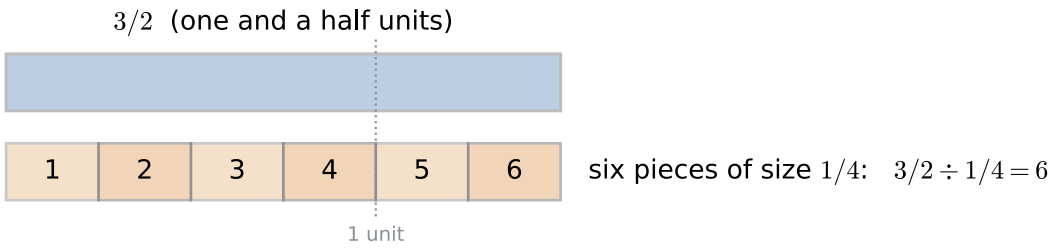


Figure 15. $\frac{3}{2} \div \frac{1}{4} = 6$: six quarter-pieces fit into one and a half.

The rule "multiply by the reciprocal" gives the same: $\frac{3}{2} \times \frac{4}{1} = \frac{12}{2} = 6$. Here are two reasons it always works.

Reason 1, from the meaning of division. $x \div y$ is the number that, multiplied by y, gives x. Check that $\frac{a}{b} \times \frac{d}{c}$ has this property with respect to $\frac{c}{d}$:

$$\left(\frac{a}{b} \times \frac{d}{c}\right) \times \frac{c}{d} = \frac{a}{b} \times \left(\frac{d}{c} \times \frac{c}{d}\right) = \frac{a}{b} \times 1 = \frac{a}{b}. \quad \checkmark$$

The middle step is the associative law; then $\frac{d}{c}$ and $\frac{c}{d}$ are reciprocals.

Reason 2, clear the inner fractions. Write the division as a fraction of fractions and multiply top and bottom by bd (which is multiplying by 1):

$$\frac{a/b}{c/d} = \frac{(a/b) \cdot bd}{(c/d) \cdot bd} = \frac{ad}{cb} = \frac{a}{b} \cdot \frac{d}{c}.$$

This second technique — multiply top and bottom by whatever clears the inner denominators — is exactly how you will simplify **complex fractions** (fractions with fractions inside) in algebra.

Example 11. Procedure — multiply and divide fractions

Input: fractions $\frac{a}{b}$, $\frac{c}{d}$ (with $c \neq 0$ for division).

Steps:

1. To divide, replace "$\div \frac{c}{d}$" by "$\times \frac{d}{c}$".
2. Cancel any factor common to a numerator and a denominator (they need not be in the same fraction).
3. Multiply numerators together and denominators together.
4. Convert mixed numbers to improper fractions *before* step 1.

Output: the product or quotient, in lowest terms.

Worked example. $2\frac{1}{4} \div 1\frac{1}{2}$.

$$
\begin{aligned}
2\tfrac{1}{4} \div 1\tfrac{1}{2} &= \frac{9}{4} \div \frac{3}{2} && \text{(mixed to improper: } 2\cdot 4+1=9,\ 1\cdot 2+1=3\text{)} \\
&= \frac{9}{4} \times \frac{2}{3} && \text{(multiply by the reciprocal)} \\
&= \frac{3\cdot 3}{2\cdot 2} \times \frac{2}{3} \\
&= \frac{3}{2} && \text{(cancel a 3 and a 2)} \\
&= 1\tfrac{1}{2}.
\end{aligned}
$$

Sanity check: $1\frac{1}{2} \times 1\frac{1}{2} = \frac{9}{4} = 2\frac{1}{4}$. ✓



Mixed numbers in algebra

$2\frac{1}{4}$ means $2 + \frac{1}{4}$, but in algebra juxtaposition means multiplication: $2x$ is $2 \times x$. So $2\frac{1}{4}$ in an algebraic context is ambiguous, and mathematicians avoid mixed numbers once letters appear, writing $\frac{9}{4}$. A second trap: multiplying a mixed number by a whole number part-by-part is fine ($2\frac{1}{4} \times 3 = 6\frac{3}{4}$, by the distributive law), but multiplying two mixed numbers part-by-part is not: $2\frac{1}{4} \times 1\frac{1}{2} \neq 2\frac{1}{8}$, because that forgets the cross terms $2 \cdot \frac{1}{2}$ and $\frac{1}{4} \cdot 1$. The true product is $\frac{9}{4} \times \frac{3}{2} = \frac{27}{8} = 3\frac{3}{8}$.



Adding tops and bottoms

$\frac{a}{b} + \frac{c}{d} \neq \frac{a+c}{b+d}$. Test: $\frac{1}{2} + \frac{1}{2} = 1$, but $\frac{1+1}{2+2} = \frac{1}{2}$. Always test a suspicious rule with simple numbers.



When adding tops and bottoms is right

The wrong-looking expression $\frac{a+c}{b+d}$ has a name — the **mediant** — and a real use. If a batter gets 3 hits in 4 at-bats one day and 1 hit in 2 the next, the overall rate is $\frac{3+1}{4+2} = \frac{4}{6}$: you combine *counts*, not rates. The mediant always lies between the two fractions. Combining rates by mediants is behind **Simpson's paradox** (Volume 3, Chapter 31), where a treatment can be better in every subgroup yet worse overall, because the subgroups have very different sizes.

2.3.5. Where you'd use this

- **Cooking.** A recipe for 4 needs $\frac{3}{4}$ cup of flour; for 6 people scale by $\frac{6}{4}$: $\frac{3}{4} \times \frac{6}{4} = \frac{18}{16} = \frac{9}{8} = 1\frac{1}{8}$ cups.
- **DIY.** How many shelves of length $1\frac{3}{4}$ ft can you cut from an 8 ft board? $8 \div \frac{7}{4} = 8 \times \frac{4}{7} = \frac{32}{7} = 4\frac{4}{7}$, so four shelves, with $8 - 4 \times \frac{7}{4} = 1$ ft left over (ignoring the saw cut).
- **Music.** Note lengths are fractions of a bar: a quarter note, two eighths, and a half note make $\frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{2} = \frac{2}{8} + \frac{1}{8} + \frac{1}{8} + \frac{4}{8} = 1$ whole bar of 4/4 time.
- **Electronics.** Resistors of 6 Ω and 3 Ω in parallel have a combined resistance R with $\frac{1}{R} = \frac{1}{6} + \frac{1}{3} = \frac{1}{6} + \frac{2}{6} = \frac{1}{2}$, so $R = 2$ Ω. Lenses obey the same "add the reciprocals" rule ($\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$).
- **Computing.** Converting CD audio (44 100 samples per second) to video audio (48 000) means resampling by the ratio $\frac{48000}{44100}$. Simplified, it is $\frac{160}{147}$ (the gcd is 300): the resampler inserts samples to multiply the rate by 160 and keeps one in 147. Exact rational arithmetic like this also appears in video frame rates (NTSC's $\frac{30000}{1001} \approx 29.97$ frames per second) and timeline editing.

2.3.6. Implementation

```python
from math import gcd

def simplify(p, q):
    """Reduce p/q to lowest terms with a positive denominator."""
    if q == 0:
        raise ZeroDivisionError("denominator is zero")
    g = gcd(p, q)                # greatest common divisor
    p, q = p // g, q // g
    if q < 0:                    # keep the sign on the numerator
        p, q = -p, -q
    return p, q

def add(a, b):
    """Add fractions a = (p1, q1) and b = (p2, q2) using the LCM of the denominators."""
    (p1, q1), (p2, q2) = a, b
    l = q1 * q2 // gcd(q1, q2)   # least common multiple
    return simplify(p1 * (l // q1) + p2 * (l // q2), l)

print(simplify(84, 126))
print(add((5, 12), (7, 18)))
print(add((1, 3), (1, 4)))
```

```
(2, 3)
(29, 36)
(7, 12)
```

In real code use `fractions.Fraction`, which does all of this (and multiplication, division, comparison) for you: `Fraction(5, 12) + Fraction(7, 18)` is `Fraction(29, 36)`.



> *Fraction workbench*
> Two fraction bars with sliders for numerator and denominator. Press "add" and the app animates re-cutting both bars into the lcm-sized pieces, then slides them together. Press "divide" and it animates tiling the first bar with copies of the second, counting the copies (including a fractional last copy).

2.3.7. Exercises

1. Compute and simplify: (a) $\frac{3}{8} + \frac{5}{12}$; (b) $\frac{7}{10} - \frac{2}{15}$; (c) $\frac{14}{15} \times \frac{25}{28}$; (d) $\frac{5}{6} \div \frac{10}{9}$.
2. Simplify the complex fraction $\dfrac{\frac{1}{2} + \frac{1}{3}}{\frac{1}{4}}$ in two ways: by computing top and bottom first, and by multiplying top and bottom by 12.
3. A recipe for 4 uses $2\frac{1}{4}$ cups of flour. How much is needed for 6?
4. Explain in words why dividing by $\frac{1}{2}$ doubles a number.

▼ **Answers**

1. (a) lcm 24: $\frac{9}{24} + \frac{10}{24} = \frac{19}{24}$. (b) lcm 30: $\frac{21}{30} - \frac{4}{30} = \frac{17}{30}$. (c) Cancel 14 with 28 (to 1 and 2) and 25 with 15 (to 5 and 3): $\frac{1 \cdot 5}{3 \cdot 2} = \frac{5}{6}$. (d) $\frac{5}{6} \times \frac{9}{10} = \frac{1 \cdot 3}{2 \cdot 2} = \frac{3}{4}$ (cancel 5 into 10, 3 into 9 and 6).
2. First way: top $\frac{3}{6} + \frac{2}{6} = \frac{5}{6}$; divided by $\frac{1}{4}$: $\frac{5}{6} \times 4 = \frac{20}{6} = \frac{10}{3}$. Second way: $$\frac{12 \cdot \frac{1}{2} + 12 \cdot \frac{1}{3}}{12 \cdot \frac{1}{4}} = \frac{6 + 4}{3} = \frac{10}{3}.$$
3. Scale by $\frac{6}{4} = \frac{3}{2}$: $\frac{9}{4} \times \frac{3}{2} = \frac{27}{8} = 3\frac{3}{8}$ cups.
4. Dividing by $\frac{1}{2}$ asks how many halves fit into the number. Every whole contains two halves, so the count is twice the number. Equivalently, multiply by the reciprocal, 2.

US curriculum

Adding and subtracting fractions with unlike denominators: **grade 5** (5.NF.1). Multiplying fractions: grade 5 (5.NF.4). Dividing a fraction by a fraction, including the "how many fit" interpretation: **grade 6** (6.NS.1). Rational expressions (the same rules with polynomials) are an **Algebra II** topic.

2.4. Decimals, and decimals that repeat

2.4.1. The problem

Every fraction is a division, and divisions can be carried out digit by digit. $\frac{1}{8} = 0.125$ stops. $\frac{1}{3} = 0.333\ldots$ never stops but repeats. $\frac{1}{7} = 0.142857142857\ldots$ repeats a block of six. What decides which happens? And going the other way, what fraction is $0.363636\ldots$?

2.4.2. Decimals are fractions in disguise

The place-value system of Section 1.3 extends to the right of the **decimal point**: the positions are worth $\frac{1}{10}$, $\frac{1}{100}$, $\frac{1}{1000}$, and so on. So

$$0.125 = \frac{1}{10} + \frac{2}{100} + \frac{5}{1000} = \frac{125}{1000} = \frac{1}{8}.$$

A terminating decimal is just a fraction whose denominator is a power of 10.



Decimal places, terminating and repeating decimals

The digits after the decimal point are the **decimal places**: the first is tenths, the second hundredths, the third thousandths. A **terminating decimal** has finitely many non-zero decimal places (0.125). A **repeating** (or **recurring**) decimal ends in a block of digits that repeats forever. The repeating block is marked with a bar (US) or dots (UK): $0.1\overline{6} = 0.1666\ldots$, $0.\overline{142857} = 0.142857142857\ldots$. The length of the repeating block is the **period**.

2.4.3. Why some fractions terminate

A fraction in lowest terms $\frac{p}{q}$ terminates exactly when it can be rewritten with denominator 10^k for some k. Since $10^k = 2^k \cdot 5^k$, that is possible precisely when **the only prime factors of q are 2 and 5**. Otherwise we would need q to divide 10^k, and a prime such as 3 or 7 never divides a power of 10.

- $\frac{3}{40}$: $40 = 2^3 \cdot 5$; terminates. Multiply top and bottom by 5^2: $\frac{75}{1000} = 0.075$.
- $\frac{7}{12}$: $12 = 2^2 \cdot 3$; the 3 forces repetition. $\frac{7}{12} = 0.58\overline{3}$.

2.4.4. Why the others repeat

Do long division of $1 \div 7$ and watch the **remainders**: 1, 3, 2, 6, 4, 5, then 1 again. Each step multiplies the remainder by 10 and divides by 7; the next digit and next remainder depend only on the current remainder. Remainders on division by 7 can only be 0 to 6. If 0 ever appears, the decimal terminates. Otherwise only six values are available, so within six steps a remainder must recur — and from then on, the whole calculation repeats exactly. That is why the period of $\frac{p}{q}$ is always less than q. *Try it:* Watch 1/7's remainders repeat .

This also proves the claim from Section 1.1: every rational number has a terminating or repeating decimal. The converse is also true (below), so irrational numbers are exactly the ones whose decimals go on forever without repeating.

PYTHON

```
def decimal_expansion(p, q):
    """Return (integer part, non-repeating digits, repeating digits) of p/q for 0 <= p, q > 0."""
    whole, r = divmod(p, q)
    digits, seen = [], {}        # seen maps remainder -> position where it first occurred
    while r != 0 and r not in seen:
        seen[r] = len(digits)
        r *= 10                  # "bring down a zero"
        d, r = divmod(r, q)
        digits.append(str(d))
    if r == 0:                   # terminating decimal
        return whole, "".join(digits), ""
    start = seen[r]              # the cycle starts where this remainder first appeared
    return whole, "".join(digits[:start]), "".join(digits[start:])

for p, q in [(1, 8), (1, 3), (5, 6), (1, 7), (3, 22)]:
    print(f"{p}/{q} ->", decimal_expansion(p, q))
```

```
1/8 -> (0, '125', '')
1/3 -> (0, '', '3')
5/6 -> (0, '8', '3')
1/7 -> (0, '', '142857')
3/22 -> (0, '1', '36')
```

This is long division, with one addition from computer science: a dictionary of previously seen remainders, which is cycle detection.

2.4.5. Converting a repeating decimal to a fraction

The trick is to shift the decimal so the repeating parts line up, then subtract to make the infinite tail vanish.

Example 12. Procedure — repeating decimal to fraction

Input: a decimal x whose repeating block has length k, starting m places after the decimal point.

Steps:

1. Multiply x by 10^m so the repeating block starts right after the point. Call this $A = 10^m x$.
2. Multiply again by 10^k to shift by one full period: $B = 10^{m+k} x$.
3. Subtract: in $B - A$ the infinite tails are identical and cancel, leaving an integer N.
4. Then $(10^{m+k} - 10^m)\, x = N$, so $x = \dfrac{N}{10^{m+k} - 10^m}$. Simplify.

Output: x as a fraction in lowest terms.

Worked example 1. $x = 0.\overline{36} = 0.363636\ldots$ ($m = 0$, $k = 2$). *Try it:* Convert 0.(36) to a fraction .

$$\begin{aligned} 100x &= 36.363636\ldots \\ x &= 0.363636\ldots \\ 100x - x &= 36 && \text{(the tails cancel)} \\ 99x &= 36 \\ x &= \frac{36}{99} = \frac{4}{11}. && \text{(divide top and bottom by 9)} \end{aligned}$$

Worked example 2. $x = 0.1\overline{6} = 0.1666\ldots$ ($m = 1$, $k = 1$). *Try it:* Convert 0.1(6) to a fraction .

$$\begin{aligned} 10x &= 1.666\ldots && \text{(repeating part now starts right after the point)} \\ 100x &= 16.666\ldots \\ 100x - 10x &= 15 \\ 90x &= 15 \\ x &= \frac{15}{90} = \frac{1}{6}. \end{aligned}$$

A shortcut falls out of example 1: a pure repeating block of length k equals the block divided by k nines. $0.\overline{7} = \frac{7}{9}$, $0.\overline{36} = \frac{36}{99}$, $0.\overline{142857} = \frac{142857}{999999} = \frac{1}{7}$.

2.4.6. Where you'd use this

- **Measurement.** Imperial drill bits and spanners come in sixty-fourths of an inch. Because $64 = 2^6$ and an inch is exactly 25.4 mm, every one of them has a terminating metric size: $\frac{5}{16}$ in $= 0.3125 \times 25.4 = 7.9375$ mm. A third of an inch has no such exact decimal.

- **Money.** US stock prices were quoted in sixteenths of a dollar until 2001; $\frac{1}{16} = \$0.0625$ terminates, but it does not fit in whole cents, which was one argument for switching to decimals. Splitting £10 three ways gives $3.\overline{3}$, so somebody must pay £3.34 (Section 2.8 shows how to do this fairly in code).
- **Computing.** The "terminates iff the denominator's primes divide the base" rule is exactly why a binary computer can store $\frac{1}{4}$ perfectly and cannot store $\frac{1}{10}$ at all (Section 2.8).

2.4.7. $0.999\ldots = 1$

Apply the shortcut to $0.\overline{9}$: it equals $\frac{9}{9} = 1$. Many people's instinct rebels: surely $0.999\ldots$ is "just a tiny bit less" than 1? Here are three arguments, from most casual to most careful.

Argument 1, from thirds. $\frac{1}{3} = 0.333\ldots$. Multiply both sides by 3: $1 = 0.999\ldots$. (If you accept the first equation, you must accept the second.)

Argument 2, the subtraction procedure.

$$\begin{aligned} x &= 0.999\ldots \\ 10x &= 9.999\ldots \\ 10x - x &= 9 \\ 9x &= 9 \\ x &= 1. \end{aligned}$$

Argument 3, there is no room between them. If $0.999\ldots$ and 1 were different numbers, there would be a gap between them, and the gap would have some positive size d. But 0.9 is within 0.1 of 1, 0.99 within 0.01, 0.999 within 0.001, and the infinite decimal is at least as close as every one of these. So d would have to be smaller than 0.1, smaller than 0.01, smaller than every 10^{-n} — and the only non-negative number smaller than all of those is 0. There is no gap, so they are the same point.

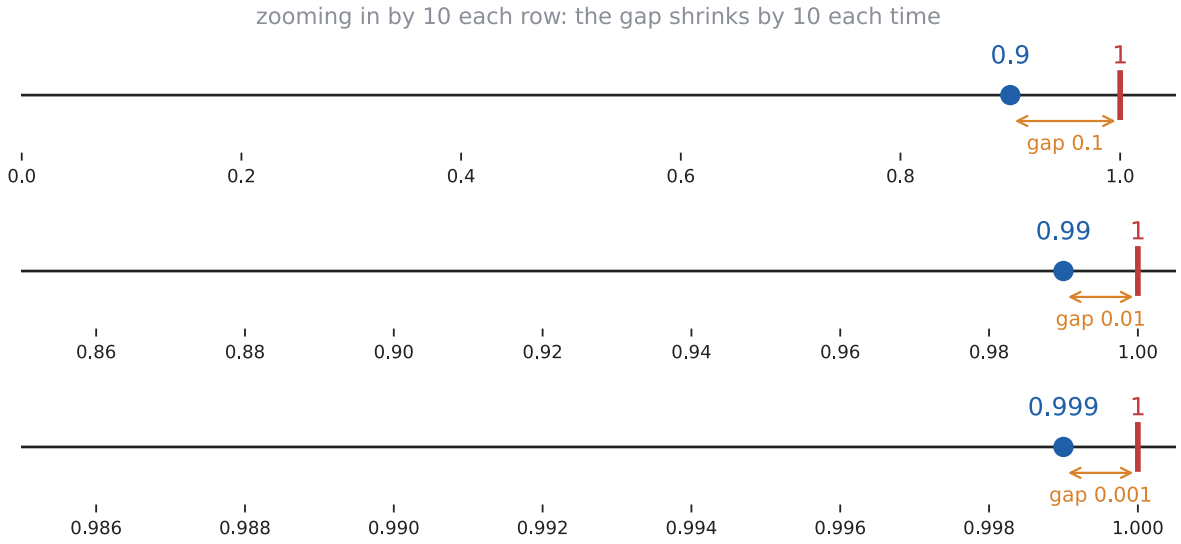


Figure 16. Each extra 9 cuts the remaining distance to 1 by a factor of ten. No positive gap survives all the steps.

The third argument is the honest one, because it says what an infinite decimal *means*: the single number that the finite truncations $0.9, 0.99, 0.999, \ldots$ approach. That is a **limit**, and making it precise is the job of Chapter 14 and Volume 2, Chapter 16 . The lesson for now: an infinite decimal is a

name for a number, and some numbers have two decimal names — $1 = 0.999\ldots$, $0.5 = 0.4999\ldots$. Only terminating decimals have this double life.



Arguing with infinity

"But it never *reaches* 1" describes the process of writing more 9s, which indeed never ends. The infinite decimal is not a process, though; it is the destination. Mixing up "the sequence $0.9, 0.99, \ldots$" (every term is less than 1) with "the limit of the sequence" (exactly 1) is the root of the confusion — and also of many genuine errors in calculus, so it is worth getting straight now.



Repeating decimal factory

Enter p and q. The app runs long division with a remainder wheel showing the values 0 to $q - 1$; each step lights up the current remainder, and when one repeats, the cycle is highlighted and the period displayed. Try every denominator from 2 to 50 and look for the pattern in which ones terminate and which have the maximum period $q - 1$ (7, 17, 19, 23, 29, 47, ...).

Try it: Open the repeating decimal factory · Repeating decimals of p/q

2.4.8. Exercises

1. Without dividing, decide which terminate: $\frac{7}{16}$, $\frac{11}{30}$, $\frac{9}{75}$, $\frac{5}{14}$.
2. Convert to fractions in lowest terms: (a) $0.\overline{45}$; (b) $0.2\overline{7}$; (c) $1.\overline{142857}$.
3. What is $0.\overline{3} + 0.\overline{6}$? Explain with fractions.
4. Explain why $\frac{1}{13}$ has a period of at most 12.

▼ **Answers**

1. $16 = 2^4$: terminates (0.4375). $\frac{11}{30}$: 30 has a factor 3, already in lowest terms: repeats ($0.3\overline{6}$). $\frac{9}{75} = \frac{3}{25}$ in lowest terms, $25 = 5^2$: terminates (0.12). Always reduce first! $\frac{5}{14}$: factor 7: repeats.
2. (a) $\frac{45}{99} = \frac{5}{11}$. (b) $10x = 2.777\ldots$, $100x = 27.777\ldots$, so $90x = 25$, $x = \frac{25}{90} = \frac{5}{18}$. (c) $1 + \frac{142857}{999999} = 1 + \frac{1}{7} = \frac{8}{7}$.
3. $\frac{1}{3} + \frac{2}{3} = 1$. So $0.333\ldots + 0.666\ldots = 0.999\ldots = 1$, consistent with Section 2.4.7.
4. The possible non-zero remainders on division by 13 are 1 to 12; within 12 steps one must repeat, and the digits repeat from there. (The actual period is 6: $\frac{1}{13} = 0.\overline{076923}$.)

US curriculum

Decimal notation for fractions: **grade 4** (4.NF.6). Converting rational numbers to decimals and knowing that they terminate or repeat: **grade 7** (7.NS.2d); converting repeating decimals to fractions: **grade 8** (8.NS.1). The $0.999\ldots = 1$ discussion is common in **Algebra II** or **Precalculus** when infinite geometric series are introduced.

2.5. Percentages

2.5.1. The problem

A share price rises 50% one year and falls 50% the next. Are you back where you started? (No — you have lost a quarter of your money.) An A/B test reports that conversion rose "from 4% to 5%, a 25% increase"; a colleague calls it "a 1% increase". Who is right? (Both, in different senses, and the sloppiness costs real decisions.) Percentages are the most widely used — and most widely misused — fractions of all.

2.5.2. Intuition and vocabulary



Percent, percentage point

Percent means "per hundred" (Latin *per centum*): $p\% = \frac{p}{100}$. So $35\% = 0.35 = \frac{7}{20}$ and $150\% = 1.5$. A **percentage point** (pp) is the unit for the *difference* between two percentages: from 4% to 5% is a rise of 1 percentage point, which is a 25% **relative** increase.

Converting is just moving the decimal point two places: to go from a decimal to a percent multiply by 100; from a percent to a decimal divide by 100.

2.5.3. The three percent questions

Every basic percentage problem is the equation

$$\text{part} = \text{rate} \times \text{whole}$$

with one of the three quantities unknown. ("Rate" is the percentage written as a decimal.)

- **Find the part:** 15% of 80 is $0.15 \times 80 = 12$.
- **Find the rate:** 12 is what percent of 80? $12/80 = 0.15 = 15\%$.
- **Find the whole:** 12 is 15% of what? $12/0.15 = 80$.

2.5.4. Percent change and multipliers

The key idea: **a percent change is a multiplication**. Increasing by $r\%$ multiplies by $1 + \frac{r}{100}$; decreasing by $r\%$ multiplies by $1 - \frac{r}{100}$. These numbers are called **multipliers** (or **growth factors**).

$$\text{percent change} = \frac{\text{new} - \text{old}}{\text{old}} \times 100\%.$$

The denominator is the **old** value: percent change is always relative to where you started.

Example 13. Procedure — successive percentage changes

Input: a starting value and a sequence of percentage changes.

Steps:

1. Convert each change to a multiplier: $+r\% \to 1 + r/100$, $-r\% \to 1 - r/100$.
2. Multiply the starting value by all the multipliers (in any order — multiplication is commutative).
3. To report the overall percent change, subtract 1 from the product of the multipliers and multiply by 100%.

Output: the final value and the overall percent change.

Worked example: the share price. Start at 100.

$$\begin{aligned} &\textbf{multipliers: } +50\% \to 1.5, \quad -50\% \to 0.5 \\ &100 \times 1.5 \times 0.5 = 100 \times 0.75 = 75 \\ &\textbf{overall change} = (0.75 - 1) \times 100\% = -25\%. \end{aligned}$$

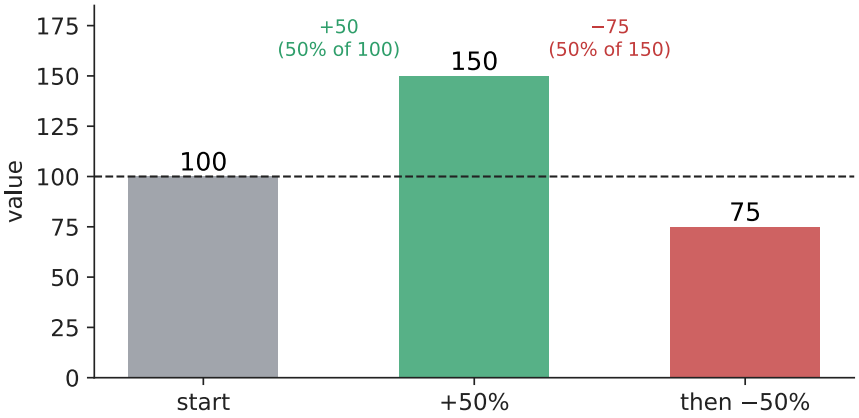


Figure 17. A 50% rise followed by a 50% fall: the fall is 50% of a larger number.

The order does not matter (0.5×1.5 is also 0.75), and *no* pair $+r\%, -r\%$ gets you back: $(1 + x)(1 - x) = 1 - x^2 < 1$ for any $x \neq 0$. To undo a 50% fall you need a 100% rise.

Worked example: reverse percentage. A price including 8% sales tax is \$54. What was the pre-tax price?

$$\begin{aligned} \textbf{pre-tax} \times 1.08 &= 54 \\ \textbf{pre-tax} &= \frac{54}{1.08} = 50. \end{aligned}$$

The tempting wrong answer, $54 - 8\% \text{ of } 54 = 49.68$, takes 8% of the *wrong whole*.

Worked example: compounding. An investment grows 10% a year for 10 years. Is that a 100% gain? The multiplier is $1.1^{10} \approx 2.594$, a gain of about 159%. Repeated multiplication by the same factor is **exponential growth**; you will meet the notation 1.1^{10} properly in Chapter 3 and the full theory in Chapter 12.

2.5.5. Choosing a method

- One change: either "find the part then add/subtract" or "multiply by the multiplier". The multiplier is one step and generalises.
- Several changes, or reversing a change: **always multipliers**. Adding and subtracting percentages is where errors live.



Percent of what?

Three common traps. (1) **Percentages do not add across changes**: $+50\%$ then -50% is -25%, not 0. (2) **Percent versus percentage points**: "unemployment rose from 4% to 6%" is a 2-point rise but a 50% relative rise; reports that do not say which are ambiguous, sometimes deliberately. (3) **Percentages of different wholes cannot be averaged directly**: a 10% discount on a $20 item and 30% on a $200 item is not a 20% discount on the basket (it is $\frac{2+60}{220} \approx 28.2\%$).

2.5.6. Where you'd use this

- **Shopping and tax.** UK VAT is 20%, so a £50 item costs £60 ($50 \times 1.2 = 60$). Going backwards, a £60 receipt contains £50 before tax ($60/1.2 = 50$) and £10 of VAT, not the £12 you get from $20\% \times 60$: the reverse-percentage trap again.
- **Health.** A food label says a serving has 6 g of saturated fat, "30% of your reference intake". The whole is $6/0.30 = 20$ g a day. A saline drip of 0.9% (weight per volume) has 0.9 g of salt per 100 ml, so a 1-litre bag holds 9 g.
- **Science.** A student measures $g = 9.6\ \text{m/s}^2$ against the accepted 9.81. The **percent error** is $\frac{|9.6-9.81|}{9.81} \times 100\% \approx 2.1\%$; it uses absolute value (Section 1.7) to report the size of the error relative to the true value.
- **Finance.** A credit card at 2% per month is not "24% a year": the yearly multiplier is $1.02^{12} \approx 1.268$, i.e. about 26.8%. Regulators require lenders to quote this compounded annual figure (the APR) for exactly that reason.
- **Computing.** A 512 GB disk with 412 GB used is $412/512 \approx 80.5\%$ full. Alert thresholds, CPU utilisation, cache hit rates, and test coverage are all "part over whole" percentages, and the sidebar below shows how changes in them get misreported.

Computer-science connection: relative change in metrics

Performance work and experimentation live on percentages, and the traps above all appear in the wild:

- "This change makes it 50% faster" is ambiguous: 50% less time (2× speed) or 50% more throughput (1.5× speed)? Prefer stating ratios: "latency went from 120 ms to 80 ms (0.67×)".
- A/B test results should report the absolute difference in percentage points and the relative lift, with an uncertainty interval (Volume 3, Chapter 31).
- Speedups compose by multiplication: three independent 10% reductions in run time give $0.9^3 = 0.729$, a 27.1% reduction, not 30%.
- Amdahl's law is a percentage argument: if 80% of a program's time can be parallelised, then even with infinitely many cores the time cannot drop below the remaining 20%, a maximum 5× speedup.



Percent playground

Stack a sequence of percentage changes as draggable blocks ($+20\%$, -15%, ...) on a starting bar. The bar updates after each block, the running multiplier is shown, and a toggle reveals the naive "add the percents" answer for comparison. Shuffle the blocks and see the final value stay the same.

2.5.7. Exercises

1. Write as a percent: $\frac{3}{8}$, 0.045, 2.5. Write as a decimal: 7%, 0.3%, 120%.
2. A laptop is reduced by 20% to \$960. What was the original price?
3. A population grows by 4% in year 1 and 6% in year 2. What is the overall percent change? Is it 10%?
4. A conversion rate goes from 2.0% to 2.6%. State the change in percentage points and as a relative percentage.

▼ **Answers**

1. 37.5%, 4.5%, 250%; 0.07, 0.003, 1.2.
2. $\text{original} \times 0.8 = 960$, so original $= 960 / 0.8 = 1200$ dollars.
3. $1.04 \times 1.06 = 1.1024$: a 10.24% increase, slightly more than 10% (the 6% in year 2 also applies to year 1's growth).
4. $+0.6$ percentage points; relative change $\frac{2.6 - 2.0}{2.0} = 0.3 = 30\%$.

US curriculum

Percent as a rate per 100: **grade 6** (6.RP.3c). Multistep percent problems (tax, tips, discounts, percent change, percent error): **grade 7** (7.RP.3). Compound growth returns in **Algebra I** (exponential functions) and **Algebra II**.

2.6. Ratios, rates, and proportion

2.6.1. The problem

A recipe for 4 people needs 300 g of flour and 2 eggs; you are cooking for 10. A screen is 1920 × 1080 pixels — what shape is it, and what size should a thumbnail be if it must be 320 pixels wide? A car uses 6.5 litres of fuel per 100 km; how far will 40 litres go? These all ask: how do quantities that change *together* relate?

2.6.2. Vocabulary and intuition



Ratio, rate, unit rate, proportion

A **ratio** compares quantities: $a : b$ (read "a to b") means "for every a of the first, there are b of the second", and it carries the same information as the fraction $\frac{a}{b}$. A **rate** is a ratio of quantities with different units: 60 km per hour, \$3 per kilogram. A **unit rate** has 1 in the second place: "per one". A **proportion** is an equation between two ratios, $\frac{a}{b} = \frac{c}{d}$. Two quantities are **directly proportional**, written $y \propto x$, if $y = kx$ for a fixed number k, the **constant of proportionality**.

A ratio can be scaled like a fraction: $300 : 2 = 150 : 1 = 750 : 5$. Scaling both parts by the same factor keeps the "for every" relationship intact — exactly like equivalent fractions.

2.6.3. The method: two ways to solve a proportion

Consider: 4 people need 300 g of flour; how much for 10?

Method 1, unit rate. Find the amount "per one", then multiply.

$$\frac{300\ \text{g}}{4\ \text{people}} = 75\ \text{g per person}, \qquad 75 \times 10 = 750\ \text{g}.$$

Method 2, scale factor. Find how many times bigger the new situation is, and scale everything by that.

$$\frac{10}{4} = 2.5, \qquad 300 \times 2.5 = 750\ \text{g}, \qquad 2 \times 2.5 = 5\ \text{eggs}.$$

Method 3, cross-multiplication. Write the proportion with an unknown and clear the denominators.

$$\begin{aligned} \frac{x}{10} &= \frac{300}{4} \\ 4x &= 300 \times 10 \qquad \text{(multiply both sides by } 4 \times 10\text{)} \\ x &= \frac{3000}{4} = 750. \end{aligned}$$

"Cross-multiplying" — the rule that $\frac{a}{b} = \frac{c}{d}$ implies $ad = bc$ — is not a separate law. It is multiplying both sides of the equation by bd, which cancels both denominators. Knowing that is what lets you use it confidently and recognise when it does not apply (it is for an equation between two single fractions, not for sums of fractions).

Example 14. Procedure — solve a direct-proportion problem

Input: a known pair (x_1, y_1) of directly proportional quantities and a new value x_2.

Steps:

1. Check that the relationship really is proportional (doubling one should double the other, and zero should give zero).
2. Compute the unit rate $k = y_1 / x_1$.
3. Compute $y_2 = k\,x_2$. (Equivalently: scale factor $s = x_2 / x_1$, then $y_2 = s\,y_1$.)
4. Check the units and that the answer moved in the expected direction.

Output: y_2.

2.6.4. Choosing a method

- Several quantities scale together (a whole recipe): **scale factor**, computed once.
- You will reuse the rate (price per kg, fuel per km): **unit rate**.
- The numbers are awkward or there is an unknown in an odd place: **cross-multiplication**, which is really the first step of solving a linear equation (Chapter 5).

2.6.5. Dividing in a ratio

Share \$240 between two people in the ratio $3 : 5$. Think in **parts**: there are $3 + 5 = 8$ equal parts, each worth $240 / 8 = 30$. So the shares are $3 \times 30 = 90$ and $5 \times 30 = 150$. Check: they sum to 240 and $90 : 150 = 3 : 5$. ✓



Not everything is proportional

If one painter takes 6 hours to paint a room, two painters do not take 12 hours: more painters means **less** time. Time and workers are **inversely proportional** ($\textbf{time} \times \textbf{workers} = \textbf{constant}$, assuming they do not get in each other's way). And many relationships are neither: a taxi fare with a fixed starting charge is not proportional to distance (zero distance does not cost zero). Check proportionality before you scale.

2.6.6. Where you'd use this

- **Maps.** On a 1 : 25 000 walking map, 4 cm on paper is $4 \times 25\,000 = 100\,000$ cm $= 1$ km on the ground.
- **Bicycles and gears.** A 48-tooth chainring driving a 16-tooth rear sprocket has gear ratio $48 : 16 = 3 : 1$: one turn of the pedals turns the wheel 3 times. With a wheel circumference of 2.1 m, each pedal stroke moves you $3 \times 2.1 = 6.3$ m.
- **Chemistry and medicine.** Diluting a stock solution 1 : 4 (one part concentrate to four parts water) gives $\frac{1}{5}$ of the original concentration, because the ratio of *parts* $1 : 4$ means 1 part in a total of 5. Confusing "1 : 4" with "$\frac{1}{4}$" is a classic dosing error.
- **Computing.** Aspect ratios (below) and resampling ratios (Section 2.3) are ratios in lowest terms. Speed is a rate: a network delivering 100 Mbit/s moves 12.5 MB per second, and a cluster processing 3000 requests per second on 4 servers runs at a unit rate of 750 per server, the number you need when deciding whether a fifth server is required.

Computer-science connection: aspect ratios

An image's **aspect ratio** is width : height in lowest terms. 1920×1080: $\gcd(1920, 1080) = 120$, so the ratio is $16 : 9$. A 320-pixel-wide thumbnail at the same aspect ratio has height $320 \times \frac{9}{16} = 180$. "Ultrawide" 2560×1080 reduces to $64 : 27$ (marketed as "21:9", which is a rounded approximation). When resizing, compute the scale factor once and apply it to both dimensions; rounding each dimension to an integer then introduces at most half a pixel of distortion. CSS's `aspect-ratio: 16 / 9` property is literally this fraction.



The golden ratio

A rectangle whose sides are in the ratio $\varphi : 1$, where $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.618$ (Greek letter phi), has the property that removing a square leaves a smaller rectangle of the same shape. Claims that the Parthenon, the Mona Lisa, or the ideal human face follow the golden ratio are mostly myth. But φ is genuinely everywhere in mathematics: ratios of consecutive Fibonacci numbers approach it (Chapter 14), and it governs the worst case of Euclid's gcd algorithm.

2.6.7. Exercises

1. Simplify the ratios $45 : 60$ and $1.5 : 2.25$.
2. A car uses 6.5 L per 100 km. How far can it go on 40 L?
3. Divide 7.2 kg of material in the ratio $1 : 2 : 3$.
4. A photo is 4000×3000 pixels. What is its aspect ratio, and what height should it have when scaled to 1200 pixels wide?

▼ **Answers**

1. $\gcd(45, 60) = 15$, so $3 : 4$. $1.5 : 2.25 = 150 : 225 = 2 : 3$ (multiply by 100, divide by 75).
2. Unit rate: $100/6.5 \approx 15.38$ km per litre. $40 \times 15.38 \approx 615$ km. (Exactly $\frac{4000}{6.5} = 615.38\ldots$ km.)
3. 6 parts, each $7.2/6 = 1.2$ kg: shares 1.2 kg, 2.4 kg, 3.6 kg.
4. $\gcd(4000, 3000) = 1000$, ratio $4 : 3$. Height $1200 \times \frac{3}{4} = 900$ pixels.

> **US curriculum**
>
> Ratios, rates, and unit rates are the core of **grade 6** (6.RP). Proportional relationships and the constant of proportionality: **grade 7** (7.RP.2). Direct and inverse variation are revisited in **Algebra I** and **Algebra II**.

2.7. Unit conversion by dimensional analysis

2.7.1. The problem

How many metres per second is 90 km/h? How long does it take to copy 500 GB over a 1 Gbit/s network link? Questions like these are where real-world arithmetic goes wrong, usually by multiplying when you should divide. **Dimensional analysis** is a mechanical method that makes such errors almost impossible: treat units as algebraic quantities that multiply and cancel just like numbers.

2.7.2. Why it works

A **conversion factor** is a fraction whose top and bottom are the same quantity in different units, such as $\frac{1000\ \text{m}}{1\ \text{km}}$. Because top and bottom are equal, the fraction equals 1, and multiplying by 1 changes nothing except the units in which the answer is expressed. Units in the numerator cancel units in the denominator exactly as common factors do in Section 2.2.



Dimensional analysis, conversion factor

Dimensional analysis (also called the **factor-label method** or **unit analysis**) converts units by multiplying by **conversion factors** — fractions equal to 1, such as $\frac{60\,\mathrm{s}}{1\,\mathrm{min}}$ — chosen so that unwanted units cancel. A **dimension** is the kind of quantity (length, time, mass); a **unit** is a particular scale for it (metre, second, kilogram). You can only add quantities with the same dimension, and both sides of a correct equation have the same dimension.

Example 15. Procedure — convert units by dimensional analysis

Input: a quantity with its units, and the target units.

Steps:

1. Write the quantity as a fraction with its units ($90\ \mathrm{km/h} = \frac{90\,\mathrm{km}}{1\,\mathrm{h}}$).
2. For each unit that must change, multiply by a conversion factor arranged so that the unwanted unit appears on the **opposite** side (top versus bottom) and cancels.
3. Cancel units; check that only the target units remain.
4. Multiply out the numbers.

Output: the quantity in the target units.

2.7.3. Worked examples

90 km/h in m/s.

$$\begin{aligned} 90\,\frac{\mathrm{km}}{\mathrm{h}} &= \frac{90\,\mathrm{km}}{1\,\mathrm{h}} \times \frac{1000\,\mathrm{m}}{1\,\mathrm{km}} \times \frac{1\,\mathrm{h}}{3600\,\mathrm{s}} \qquad \text{(km on top cancels km on bottom; h likewise)} \\ &= \frac{90 \times 1000}{3600}\,\frac{\mathrm{m}}{\mathrm{s}} \\ &= 25\,\mathrm{m/s}. \end{aligned}$$

Copying 500 GB over a 1 Gbit/s link (ignoring protocol overhead; here G means 10^9 and a byte is 8 bits).

$$\begin{aligned} 500\,\mathrm{GB} \times \frac{8\,\mathrm{Gbit}}{1\,\mathrm{GB}} \times \frac{1\,\mathrm{s}}{1\,\mathrm{Gbit}} &= 4000\,\mathrm{s} \\ 4000\,\mathrm{s} \times \frac{1\,\mathrm{min}}{60\,\mathrm{s}} &\approx 66.7\,\mathrm{min}. \end{aligned}$$

Notice the rate "1 Gbit per second" was used upside down, as $\frac{1\,\mathrm{s}}{1\,\mathrm{Gbit}}$, because we wanted seconds on top. A rate is a conversion factor; you can flip it whenever the units call for it.

2.7.4. Failure modes and method choice

Dimensional analysis cannot tell you whether a conversion factor is correct (is a mile 1.609 km or 1.069 km?), and it cannot catch errors between units of the same dimension that are not written down. Its power is that wrong **arrangements** of correct factors leave visibly wrong units — if your "speed" comes out in $\mathrm{km \cdot h}$, you multiplied where you should have divided. For one-step conversions people often just "know" whether to multiply or divide; for anything with two or more steps, write out the factors.



> *The $125 million unit error*
>
> In 1999 NASA's Mars Climate Orbiter burned up in the Martian atmosphere because one team's software reported thruster impulse in pound-force seconds while another team's software expected newton-seconds — a factor of about 4.45. Every number was computed correctly; the units were never checked. Modern physics libraries and languages such as F# carry units in the type system so that this kind of mismatch is a compile error.

Computer-science connection: units in code

Most production bugs involving units are of three kinds: **time** (milliseconds versus seconds versus nanoseconds — Unix timestamps are seconds, JavaScript's `Date.now()` is milliseconds), **size** (bits versus bytes, and 10^3 versus 2^{10} multipliers — see Chapter 3), and **money** (dollars versus cents). Defences: put the unit in the name (`timeout_ms`, `size_bytes`), use typed durations (`datetime.timedelta`, `std::chrono`), and convert at system boundaries, never in the middle of a calculation.

2.7.5. Exercises

1. Convert 1 mile = 1.609 km to express 60 mph in km/h and in m/s.
2. A disk sustains 200 MB/s. How many hours to read 3 TB? (Use $1\ \mathrm{TB} = 10^6\ \mathrm{MB}$.)
3. A medication dose is 15 mg per kg of body mass per day, split into 3 equal doses. What is each dose for a 22 kg child?

▼ **Answers**

1. $60\,\frac{\mathrm{mi}}{\mathrm{h}} \times \frac{1.609\ \mathrm{km}}{1\ \mathrm{mi}} = 96.54$ km/h. Then $96.54\,\frac{\mathrm{km}}{\mathrm{h}} \times \frac{1000\ \mathrm{m}}{1\ \mathrm{km}} \times \frac{1\ \mathrm{h}}{3600\ \mathrm{s}} \approx 26.8$ m/s.
2. $3 \times 10^6\ \mathrm{MB} \times \frac{1\ \mathrm{s}}{200\ \mathrm{MB}} = 15\,000$ s; $15\,000\ \mathrm{s} \times \frac{1\ \mathrm{h}}{3600\ \mathrm{s}} \approx 4.17$ h (about 4 h 10 min).
3. $22\ \mathrm{kg} \times \frac{15\ \mathrm{mg}}{1\ \mathrm{kg \cdot day}} = 330$ mg per day; $330/3 = 110$ mg per dose.

US curriculum

Converting measurement units using ratio reasoning: **grade 6** (6.RP.3d). Dimensional analysis as a named technique is mostly taught in high-school **chemistry** and **physics**; **Algebra I** standards (N-Q.1) ask students to "use units as a way to understand problems".

2.8. Fractions inside a computer: why 0.1 is not 0.1

2.8.1. The problem

```
>>> 0.1 + 0.2
0.30000000000000004
>>> 0.1 + 0.2 == 0.3
False
>>> sum([0.1] * 10)
0.9999999999999999
```

PYTHON

Try it: Why 0.1 × 10 ≠ 1 .

This is not a bug in Python. Every language using standard floating point behaves identically. With what you now know about terminating decimals, you can explain exactly why.

2.8.2. Why it happens

In Section 2.4 we saw that a fraction in lowest terms terminates in base 10 exactly when its denominator has no prime factors other than 2 and 5 — the prime factors of 10. The same argument works in any base: **a fraction terminates in base 2 exactly when its denominator is a power of 2.**

$0.1 = \frac{1}{10}$ has a factor 5 in its denominator. So in binary it repeats forever, just as $\frac{1}{3}$ does in decimal:

$$0.1_{10} = 0.0\overline{0011}_2 = 0.0001100110011001100110011\ldots_2$$

A `double` stores 53 significant bits, so it must cut this infinite expansion off and round. The value actually stored for `0.1` is the nearest number of the form $\frac{\text{integer}}{2^k}$:

```
>>> from decimal import Decimal
>>> Decimal(0.1)
Decimal('0.1000000000000000055511151231257827021181583404541015625')
>>> (0.1).as_integer_ratio()
(3602879701896397, 36028797018963968)
```

PYTHON

Try it: See how 0.1 is stored .

The denominator is 2^{55}. Similarly `0.2` and `0.3` are stored slightly off, the errors do not cancel exactly, and the rounded sum of the stored `0.1` and `0.2` is a different double from the stored `0.3`.
Try it: How 0.3 is stored ; How 0.2 is stored .

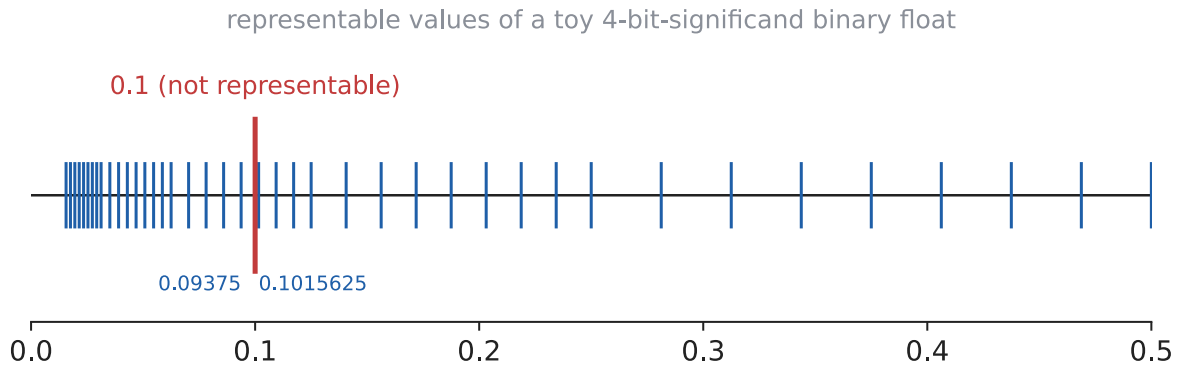


*Figure 18. Floating-point numbers are a finite set of binary fractions, spaced more widely as they grow. **0.1** falls between two of them. (A real `double` has 53 significant bits rather than this toy's 4, so its grid is unimaginably finer — but 0.1 still falls between the points.)*

Floating point

A **floating-point number** is stored as a sign, a **significand** (or **mantissa**) of fixed length, and an **exponent**: in binary, $\pm 1.b_1 b_2 \ldots b_{52} \times 2^e$ for an IEEE 754 `double`. It is binary scientific notation (Chapter 3). Only numbers that are integers divided by a power of 2 can be represented exactly, and only if they fit in the available bits. The gap between neighbouring floats grows with their size; relative precision is roughly 16 significant decimal digits.

2.8.3. What to do about it: the method

There is no single fix; there is a choice of representation, made by asking what the numbers mean.

Example 16. Procedure — choose a number representation for fractional values

Input: a quantity your program must store and compute with.

Steps:

1. If the quantity is **money** or any count of a smallest unit: store an **integer** number of the smallest unit (cents, satoshis, milliseconds). Integer arithmetic is exact. Round only when dividing (interest, splitting a bill), using an explicit rule.
2. If you need exact decimal fractions with a fixed number of places (accounting, tax law): use a **decimal** type (`decimal.Decimal` in Python, `BigDecimal` in Java, `NUMERIC` in SQL).
3. If you need exact rational arithmetic (symbolic math, exact geometry): use a **rational** type (`fractions.Fraction`) and accept that numerators and denominators may grow large.
4. Otherwise (measurements, physics, graphics, machine learning): use **floating point**, and never test for exact equality — compare with a tolerance (Section 1.7), e.g. `math.isclose(a, b)`.

Output: a representation whose errors are either zero or understood.

Worked example: splitting a bill in integer cents. $100.00 split three ways. In cents, $10\,000 = 3 \times 3333 + 1$. Two people pay 3333 cents ($33.33) and one pays 3334 cents ($33.34); total exactly 10,000. With floats, `100/3` gives `33.333333333333336`, and rounding each share to cents gives three times $33.33 = $99.99 — a cent has vanished. Division with remainder (Section 1.8) is the right tool, and it forces you to decide who gets the extra cent.

```python
from fractions import Fraction
from decimal import Decimal

print(Fraction(1, 10) + Fraction(2, 10) == Fraction(3, 10))   # exact rationals
print(Decimal("0.1") + Decimal("0.2") == Decimal("0.3"))      # exact decimals
print(10 + 20 == 30)                                          # integer cents
cents, n = 10_000, 3
q, r = divmod(cents, n)
print([q + 1 if i < r else q for i in range(n)])              # fair split
```

```
True
True
True
[3334, 3333, 3333]
```



`Decimal(0.1)` *is not* `Decimal("0.1")`

Constructing a decimal or rational from a float captures the float's error: `Decimal(0.1)` is the 55-digit monster above, and `Fraction(0.1)` is $\frac{3602879701896397}{36028797018963968}$. Construct from strings or integers instead: `Decimal("0.1")`, `Fraction(1, 10)`. Likewise, parse money from text directly into integer cents; never via `float`.

Computer-science connection: fixed point

Fixed-point arithmetic stores a real number as an integer with an implied denominator: in "16.16 fixed point", the integer n represents $n/2^{16}$. Integer cents are fixed point with denominator 100. Fixed point was standard in games and DSP before floating-point hardware was cheap (the original *Doom* used 16.16), and it is back in machine learning, where neural network weights are **quantised** to 8-bit or 4-bit integers with a scale factor — a fraction $\frac{n}{\text{scale}}$ again. Every one of these is the fraction concept from this chapter, chosen deliberately for speed or exactness.

The Patriot missile clock

In 1991 a Patriot missile battery in Dhahran failed to intercept an incoming Scud missile; 28 soldiers died. The system counted time in tenths of a second and converted to seconds by multiplying by 0.1, stored in a 24-bit fixed-point register — so it was chopped to a binary fraction just under 0.1, off by about 9.5×10^{-8}. After 100 hours of continuous operation, that tiny error had accumulated to about 0.34 seconds, enough for the tracking window to miss a target moving at over 1,600 m/s.

Float explorer

Type a decimal number and see its exact stored binary value, its sign/exponent/significand bits, the neighbouring representable doubles on either side, and the error. A second panel sums 0.1 a chosen number of times as float, as `Decimal`, and as integer tenths, plotting the float's drift.

Try it: Watch 0.1 drift · Float explorer: drift of 0.1 · The same drift in float32 · 0.25 is stored exactly

2.8.4. Exercises

1. Which of these are exactly representable as binary floating point (given enough bits)? 0.5, 0.75, 0.2, $\frac{3}{64}$, $\frac{1}{3}$.
2. Write 0.375 in binary.
3. Explain why `sum([0.1]*10) == 1.0` is `False` but `sum([0.5]*10) == 5.0` is `True`.

▼ **Answers**

1. $0.5 = \frac{1}{2}$, $0.75 = \frac{3}{4}$, and $\frac{3}{64}$ have power-of-two denominators: exact. $0.2 = \frac{1}{5}$ and $\frac{1}{3}$: not exact.
2. $0.375 = \frac{3}{8} = \frac{1}{4} + \frac{1}{8} = 0.011_2$.
3. 0.5 is stored exactly and every partial sum $0.5, 1.0, \ldots, 5.0$ is exactly representable, so no rounding ever occurs. 0.1 is stored with a small error, and each addition may round again; the errors accumulate to give 0.9999999999999999.

US curriculum

Floating-point representation is not part of US school mathematics; it appears in **AP Computer Science A** (as `double` precision issues) and in college **computer organisation** and **numerical methods** courses. The underlying mathematical fact — which fractions terminate in a given base — extends grade 7–8 work on terminating and repeating decimals.

2.9. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Number theory step by step
- Number theory
- Floating-point inspector
- Where floating point bites

2.10. Mastery check

1. Compute, simplifying fully: $\frac{5}{6} - \frac{3}{8} + \frac{1}{4}$.
2. Compute $1\frac{2}{3} \div \frac{5}{9}$, and explain in one sentence why dividing by a fraction is multiplying by its reciprocal.
3. Convert $0.3\overline{18}$ to a fraction in lowest terms. Will $\frac{7}{24}$ terminate as a decimal? Why or why not?
4. Give a short argument that $0.999\ldots = 1$ which does not rely on manipulating infinite decimals with the subtraction trick.
5. A price rises 25% then falls 20%. What is the overall change? An item costs $69 after a 15% discount; what was the original price?
6. Share 84 in the ratio $2:5$. A 3840 × 2160 video is scaled to 1280 wide; what is its height, and what is its aspect ratio?
7. Convert 5 m/s to km/h using dimensional analysis.
8. Explain why 0.25 is stored exactly as a double but 0.3 is not, and say how you would store a bank balance.

Decision rule: All correct without hesitation → move on to Chapter 3. Stuck on 1–2 → reread the matching section (Section 2.3, Section 2.4, Section 2.5, Section 2.6, Section 2.7, Section 2.8). Stuck on more → work the section exercises first; fractions are the most common hidden weakness under later algebra, so the time is well spent.

▼ **Mastery check answers**

1. lcm $(6, 8, 4) = 24$: $\frac{20}{24} - \frac{9}{24} + \frac{6}{24} = \frac{17}{24}$.
2. $\frac{5}{3} \times \frac{9}{5} = \frac{9}{3} = 3$. Dividing by $\frac{c}{d}$ asks for the number that multiplied by $\frac{c}{d}$ gives the original, and $\frac{a}{b} \cdot \frac{d}{c}$ is that number since $\frac{d}{c} \cdot \frac{c}{d} = 1$.
3. $x = 0.3181818\ldots$; $10x = 3.181818\ldots$; $1000x = 318.181818\ldots$; subtract: $990x = 315$, $x = \frac{315}{990} = \frac{7}{22}$. $\frac{7}{24}$: $24 = 2^3 \cdot 3$ and 7/24 is in lowest terms, so the factor 3 makes it repeat ($0.29\overline{16}$).
4. If they differed, the gap would be a positive number. But $0.999\ldots$ is within 10^{-n} of 1 for every n, so the gap is smaller than every 10^{-n}; the only non-negative number with that property is 0. (Or: $\frac{1}{3} = 0.\overline{3}$, times 3.)

5. $1.25 \times 0.8 = 1.00$: no overall change. Original $= 69 / 0.85 = 81.18$ dollars (to the cent; $0.85 \times 81.18 = 69.003$).
6. 7 parts of 12: shares 24 and 60. Scale factor $\frac{1280}{3840} = \frac{1}{3}$: height 720; aspect ratio $16:9$ ($\gcd(3840, 2160) = 240$).
7. $5 \frac{\text{m}}{\text{s}} \times \frac{1\,\text{km}}{1000\,\text{m}} \times \frac{3600\,\text{s}}{1\,\text{h}} = 18$ km/h.
8. $0.25 = \frac{1}{4}$ has a power-of-two denominator, so its binary expansion 0.01_2 terminates. $0.3 = \frac{3}{10}$ has a factor 5 in the denominator, so it repeats in binary and must be rounded. Store money as an integer number of cents (or with a decimal type), never as a float.

Chapter 3. Exponents, Roots, and Scientific Notation

Chapter map: an exponent is shorthand for repeated multiplication, and from that one idea everything else in this chapter is derived rather than memorised: the laws of exponents, why $a^0 = 1$, why a negative exponent means a reciprocal, why a fractional exponent means a root. We then put powers to work — simplifying surds, writing very large and very small numbers in scientific notation, reasoning in orders of magnitude, and living in the powers-of-two world of computer memory. We finish with a first look at exponential growth and its inverse, the logarithm, and with one of the most elegant algorithms in computing: exponentiation by squaring.

Why it matters: powers are the language of scale. Compound interest, half-lives of medicines, the inverse-square law of light and gravity, the size of a pizza, how many passwords an attacker must try, how much memory a 64-bit pointer can address, and why some algorithms finish in a blink while others never finish at all are all questions about exponents.

Where this chapter sits

Prerequisites: Chapter 1 (arithmetic laws, negative numbers, binary) and Chapter 2 (fractions, reciprocals, decimals).

This chapter unlocks: algebraic expressions and polynomials (Chapter 4, Chapter 7), the quadratic formula's square root (Chapter 8), exponential functions and logarithms (Chapter 12), and the power rule of calculus (Volume 2, Chapter 17). Fast exponentiation reappears in number theory and cryptography (Volume 3, Chapter 23).

Fast path: if the mastery check is quick, move on — but skim Section 3.3 (why the rules are what they are) and Section 3.9 (the algorithm).

3.1. Exponents as repeated multiplication

3.1.1. The problem

Take a sheet of paper 0.1 mm thick and fold it in half. It is now 0.2 mm thick. Fold again: 0.4 mm. How many folds until the stack would reach the Moon, 384,000 km away? Most people guess thousands, or millions. The answer is 42. Repeated doubling grows so violently that ordinary intuition, trained on repeated *adding*, fails completely. We need notation for repeated multiplication, and then we need to learn to think with it.

3.1.2. Notation and vocabulary

Just as multiplication abbreviates repeated addition ($3 \times 4 = 4 + 4 + 4$), an **exponent** abbreviates repeated multiplication:

$$a^n = \underbrace{a \times a \times \cdots \times a}_{n \text{ factors}}.$$

The underbrace says "count the factors underneath: there are n of them". So $2^5 = 2 \times 2 \times 2 \times 2 \times 2 = 32$, and after n folds the paper is 0.1×2^n mm thick.



Base, exponent, power

In a^n, a is the **base** and n is the **exponent** (also called the **index** — plural **indices** — or **power**). The whole expression a^n is "a **power** of a" and is read "a to the n", "a to the power n", or "a raised to the nth". $a^1 = a$. In code it is written `a**n` (Python), `a^n` (spreadsheets, math software), or `pow(a, n)` in C; in C and Java `^` means XOR.



Squared, cubed

a^2 is read "a **squared**" because it is the area of a square with side a; a^3 is "a **cubed**", the volume of a cube with edge a. A **perfect square** is the square of an integer (1, 4, 9, 16, 25, …); a **perfect cube** is the cube of one (1, 8, 27, 64, …).

Back to the paper. After 42 folds the thickness is 0.1×2^{42} mm. We will see in Section 3.7 that $2^{10} \approx 1000$, so $2^{40} = (2^{10})^4 \approx 10^{12}$ and $2^{42} \approx 4 \times 10^{12}$. That makes $0.1 \times 4 \times 10^{12} = 4 \times 10^{11}$ mm $= 4 \times 10^5$ km: about 400,000 km, beyond the Moon. (Physically you cannot fold paper more than about 7 times — the point is the arithmetic.)

3.1.3. Failure modes



Exponent mistakes worth avoiding from day one

- $2^3 = 8$, not $2 \times 3 = 6$. The exponent counts factors; it does not multiply the base.
- $-3^2 = -9$ but $(-3)^2 = 9$: the exponent binds tighter than the leading minus (Section 1.6). An exponent applies only to what is immediately beneath it: $2x^2$ means $2 \cdot x^2$, not $(2x)^2$.
- An even power of a negative number is positive; an odd power is negative: $(-2)^4 = 16$, $(-2)^5 = -32$ (count the negative factors, Section 1.5).

US curriculum

Whole-number exponents, especially powers of 10, appear in **grade 5–6** (5.NBT.2, 6.EE.1). Integer exponents and their properties are **grade 8** (8.EE.1). Rational exponents and radicals are **Algebra I** / **Algebra II** (N-RN.1–2).

3.2. The laws of exponents, derived

3.2.1. The problem

Simplify $\dfrac{x^7 \cdot (x^2)^3}{x^4}$. You might remember rules like "add the exponents" and "multiply the exponents", but which applies when? If the rules are derived rather than memorised, you never need to remember them: you can re-derive any of them in ten seconds by counting factors.

3.2.2. Why they work: counting factors

Product law. $a^m \cdot a^n$: write out the factors.

$$a^m \cdot a^n = \underbrace{a \cdots a}_{m} \cdot \underbrace{a \cdots a}_{n} = \underbrace{a \cdots a}_{m+n} = a^{m+n}.$$

Example: $2^3 \cdot 2^2 = (2 \cdot 2 \cdot 2)(2 \cdot 2) = 2^5$. The exponents add because the factors are just pooled (associative law).

Quotient law. $\dfrac{a^m}{a^n}$ with $m > n$ and $a \neq 0$: each of the n factors downstairs cancels one of the m factors upstairs.

$$\frac{a^m}{a^n} = \frac{\overbrace{a \cdots a}^{m}}{\underbrace{a \cdots a}_{n}} = \underbrace{a \cdots a}_{m-n} = a^{m-n}.$$

Example: $\dfrac{5^6}{5^2} = \dfrac{5 \cdot 5 \cdot 5 \cdot 5 \cdot (5 \cdot 5)}{(5 \cdot 5)} = 5^4$.

Power of a power. $(a^m)^n$ is n copies of a^m multiplied together, and each copy contains m factors of a:

$$(a^m)^n = \underbrace{a^m \cdot a^m \cdots a^m}_{n \text{ copies}} = a^{\overbrace{m+m+\cdots+m}^{n}} = a^{mn}.$$

Example: $(2^3)^2 = 2^3 \cdot 2^3 = 2^6$.

Power of a product and **power of a quotient.** Rearranging factors (commutative and associative laws):

$$(ab)^n = \underbrace{(ab)(ab)\cdots(ab)}_{n} = \underbrace{a \cdots a}_{n}\underbrace{b \cdots b}_{n} = a^n b^n, \qquad \left(\frac{a}{b}\right)^n = \frac{a^n}{b^n}.$$

Collected:

$$\begin{aligned}
a^m \cdot a^n &= a^{m+n} && \text{(same base, multiply } \to \text{ add exponents)} \\
\frac{a^m}{a^n} &= a^{m-n} && \text{(same base, divide } \to \text{ subtract exponents)} \\
(a^m)^n &= a^{mn} && \text{(power of a power } \to \text{ multiply exponents)} \\
(ab)^n &= a^n b^n && \text{(power distributes over a product)} \\
\left(\frac{a}{b}\right)^n &= \frac{a^n}{b^n} && \text{(power distributes over a quotient)}
\end{aligned}$$

Laws of exponents

For non-zero bases and integer exponents: $a^m a^n = a^{m+n}$, $a^m / a^n = a^{m-n}$, $(a^m)^n = a^{mn}$, $(ab)^n = a^n b^n$, $(a/b)^n = a^n / b^n$, $a^0 = 1$, $a^{-n} = 1/a^n$. They extend to fractional exponents for **positive** bases. Each can be re-derived by writing out the factors.

3.2.3. The method

Example 17. Procedure — simplify an expression with exponents

Input: a product/quotient of powers.

Steps:

1. Expand any power of a product or quotient: $(ab)^n \to a^n b^n$.
2. Apply power-of-a-power: $(a^m)^n \to a^{mn}$.
3. For each base, combine everything into a single power: add exponents of factors on top, subtract exponents of factors on the bottom.
4. Rewrite any negative exponents as reciprocals if the answer should have positive exponents (Section 3.3).
5. Evaluate numerical powers if small.

Output: a product of powers of distinct bases.

3.2.4. Worked examples

Example 1. The opening problem.

$$\begin{aligned} \frac{x^7 \cdot (x^2)^3}{x^4} &= \frac{x^7 \cdot x^6}{x^4} && \text{(power of a power: } 2 \times 3 = 6\text{)} \\ &= \frac{x^{13}}{x^4} && \text{(product law: } 7 + 6 = 13\text{)} \\ &= x^9. && \text{(quotient law: } 13 - 4 = 9\text{)} \end{aligned}$$

Example 2. Several bases.

$$\begin{aligned} \frac{(2a^3 b)^2 \cdot 3ab^4}{6a^5 b^2} &= \frac{2^2 a^6 b^2 \cdot 3ab^4}{6a^5 b^2} && \text{(power of a product, then power of a power)} \\ &= \frac{12\, a^7 b^6}{6\, a^5 b^2} && (4 \times 3 = 12;\ a^6 a^1 = a^7;\ b^2 b^4 = b^6) \\ &= 2a^2 b^4. && (12/6;\ 7 - 5;\ 6 - 2) \end{aligned}$$

Example 3. Different bases that are secretly the same: $8^3 \cdot 4^2$.

$$8^3 \cdot 4^2 = (2^3)^3 \cdot (2^2)^2 = 2^9 \cdot 2^4 = 2^{13} = 8192.$$

Rewriting numbers as powers of a common base is the standard trick for exponential equations in Chapter 12.

3.2.5. Failure modes



Things the laws do not say

- **No law for sums.** $a^m + a^n$ does not simplify to a single power: $2^3 + 2^2 = 12$, which is not a power of 2. The laws are about products.
- **Powers do not distribute over addition.** $(a+b)^2 \neq a^2 + b^2$. Test: $(1+1)^2 = 4$ but $1^2 + 1^2 = 2$. The correct expansion, $a^2 + 2ab + b^2$, comes from the distributive law (Chapter 7).
- **Different bases do not combine by adding exponents.** $2^3 \cdot 3^2 \neq 6^5$. (But equal exponents do combine: $2^3 \cdot 3^3 = 6^3$, by the power-of-a-product law read backwards.)
- $(a^m)^n$ is not a^{m^n}: $(2^3)^2 = 2^6 = 64$ while $2^{3^2} = 2^9 = 512$ (stacked exponents group from the top, Section 1.6).

3.2.6. Where you'd use this

- **Area and volume scale as powers.** A 16-inch pizza versus a 12-inch one: the area ratio is $(16/12)^2 = (4/3)^2 = 16/9 \approx 1.78$, so the larger pizza is almost 80% more food, not 33% more. Double every edge of a box and its volume grows by $2^3 = 8$. This is the **square–cube law**: a giant ant scaled up 10 times would weigh $10^3 = 1000$ times as much but have legs only $10^2 = 100$ times as strong in cross-section, which is why large animals have thick legs.
- **Unit conversions with powers.** $1\ \text{km}^2 = (10^3\ \text{m})^2 = 10^6\ \text{m}^2$ (power of a product, then power of a power). Forgetting to square the conversion factor is one of the commonest errors in physics homework and in real engineering.
- **Counting in computing.** A pixel with 8 bits each for red, green, and blue can be one of $2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16\,777\,216$ colours (product law). A 4-digit PIN has 10^4 possibilities; a 6-digit one, 10^6, which is 100 times harder to guess.



Exponent rule checker

Enter an exponent "rule" — true or false — such as $(a+b)^n = a^n + b^n$ or $a^m a^n = a^{mn}$. The app tests it on random numbers and shows a counterexample if one exists, or a factor-counting picture (rows of coloured tiles for each factor) if it holds.

3.2.7. Exercises

1. Simplify: (a) $x^4 \cdot x^5$; (b) $\dfrac{y^{10}}{y^3}$; (c) $(z^3)^4$; (d) $(3p^2q)^3$.
2. Simplify $\dfrac{(4m^2n^3)^2}{8m^3n^5}$.

3. Write $27^2 \cdot 9^3 / 3^5$ as a single power of 3.
4. Is $2^{10} + 2^{10} = 2^{11}$ or 2^{20}? Explain.

▼ **Answers**

1. (a) x^9. (b) y^7. (c) z^{12}. (d) $27p^6q^3$.
2. Numerator $16m^4n^6$; divide: $\frac{16}{8} m^{4-3} n^{6-5} = 2mn$.
3. $(3^3)^2 \cdot (3^2)^3 / 3^5 = 3^6 \cdot 3^6 / 3^5 = 3^7$.
4. $2^{10} + 2^{10} = 2 \cdot 2^{10} = 2^1 \cdot 2^{10} = 2^{11}$. Adding two equal things doubles; it does not multiply the exponents.

3.3. Zero and negative exponents

3.3.1. The problem

"a^n means n factors of a" makes sense for $n = 1, 2, 3, \ldots$. But what could 2^0 mean — zero factors of 2? Or 2^{-3}, negative three factors? These are not questions about the definition (which has nothing to say) but about how to *extend* it. We want an extension that keeps the laws of exponents working. It turns out there is exactly one.

3.3.2. Intuition: follow the pattern

Read the powers of 2 from right to left: each step to the left divides by 2.

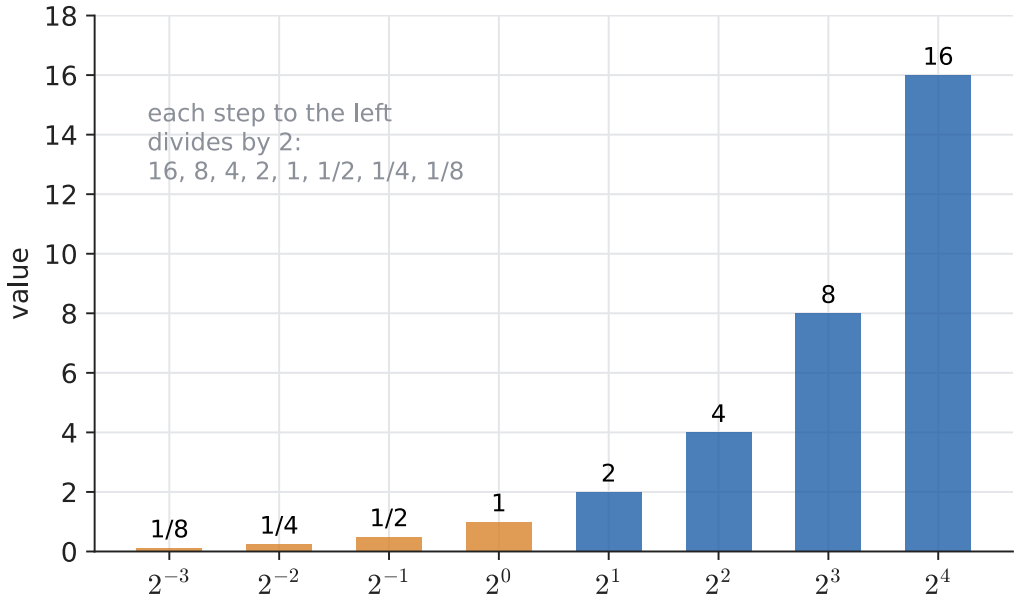


Figure 19. Powers of 2. Moving one step left always divides by 2; continuing the pattern past 2^1 gives $2^0 = 1$, then halves.

$$
\begin{aligned}
2^3 &= 8 \\
2^2 &= 4 \qquad (\div 2) \\
2^1 &= 2 \qquad (\div 2) \\
2^0 &= 1 \qquad (\div 2) \\
2^{-1} &= \tfrac{1}{2} \qquad (\div 2) \\
2^{-2} &= \tfrac{1}{4} \qquad (\div 2) \\
2^{-3} &= \tfrac{1}{8}. \qquad (\div 2)
\end{aligned}
$$

If the pattern is to continue — and there is no reason to put a break in it — then 2^0 must be 1 and 2^{-n} must be $\frac{1}{2^n}$.

3.3.3. Why it works: the laws force it

Insist that the quotient law $\frac{a^m}{a^n} = a^{m-n}$ keeps working when $m = n$. The left side is something divided by itself:

$$1 = \frac{a^n}{a^n} = a^{n-n} = a^0.$$

So $a^0 = 1$ for every $a \neq 0$. Now insist that the product law keeps working for a negative exponent:

$$a^{-n} \cdot a^n = a^{-n+n} = a^0 = 1.$$

So a^{-n} is the number which, multiplied by a^n, gives 1 — the reciprocal of a^n:

$$a^{-n} = \frac{1}{a^n}, \qquad a^0 = 1 \qquad (a \neq 0).$$

This is a recurring pattern in mathematics: a definition is first made where it obviously makes sense, then **extended** in the unique way that preserves the rules. You will see it again with fractional exponents (next section), with $0! = 1$ (Volume 3, Chapter 24), and with complex exponents (Chapter 15).



Zero and negative exponents

For any $a \neq 0$: $a^0 = 1$ (the empty product), and $a^{-n} = \dfrac{1}{a^n}$. A negative exponent means "reciprocal", not "negative": $2^{-3} = \frac{1}{8}$, which is positive. A negative exponent on a fraction flips it: $\left(\frac{a}{b}\right)^{-n} = \left(\frac{b}{a}\right)^{n}$.

3.3.4. Worked examples

$$5^{-2} = \frac{1}{5^2} = \frac{1}{25} = 0.04$$

$$\left(\tfrac{2}{3}\right)^{-3} = \left(\tfrac{3}{2}\right)^{3} = \frac{27}{8}$$

$$\frac{x^3}{x^8} = x^{3-8} = x^{-5} = \frac{1}{x^5}$$

$$\frac{6a^{-2}b^3}{2a^3b^{-1}} = 3\,a^{-2-3}\,b^{3-(-1)} = 3a^{-5}b^4 = \frac{3b^4}{a^5}.$$

In the last line, note $b^{3-(-1)} = b^4$: subtracting a negative exponent is adding (Section 1.5). An equivalent shortcut is "a factor with a negative exponent may move across the fraction bar and change the sign of its exponent".



The empty product and 0^0

$a^0 = 1$ for the same reason an empty sum is 0: 1 is what you start with before multiplying anything in (it is the multiplicative identity, Section 1.4). In code, `product = 1` before a loop that runs zero times. What about 0^0? The pattern $0^1, 0^2, \ldots = 0$ suggests 0, while $a^0 = 1$ suggests 1. In algebra, combinatorics, and computing the convention $0^0 = 1$ is almost universal (Python's `0**0` is `1`), because it makes formulas like the binomial theorem work. In calculus, "0^0" is an **indeterminate form** (Volume 2, Chapter 16): a limit of that shape can come out to anything.



0^{-1}

0^{-n} for $n > 0$ would be $\frac{1}{0^n} = \frac{1}{0}$, which is undefined (Section 1.4). Python raises `ZeroDivisionError` for `0**-1`. The laws of this section all assume a non-zero base.

3.3.5. Where you'd use this

- **Medicine.** Caffeine has a half-life of roughly 5 hours. After n half-lives a fraction 2^{-n} remains: 200 mg drunk at 2 p.m. leaves about $200 \times 2^{-2} = 50$ mg in your system at midnight. Drug dosing schedules are designed with exactly this calculation.

- **Physics: inverse-square laws.** Light intensity, gravity, and radio signal strength fall off as r^{-2}. Move three times further from a lamp (or a Wi-Fi router) and you receive $3^{-2} = \frac{1}{9}$ of the power. Units use negative exponents too: acceleration in $\mathrm{m\,s^{-2}}$, density in $\mathrm{kg\,m^{-3}}$.
- **Computing.** Binary place values to the right of the point are $2^{-1}, 2^{-2}, 2^{-3}, \ldots$ (Section 2.8). Machine-learning learning rates, tolerances, and probabilities are routinely written as 10^{-3} or `1e-3`.

3.3.6. Exercises

1. Evaluate: 10^{-3}, 4^{-1}, $(-3)^{-2}$, $\left(\frac{1}{2}\right)^{-4}$, 7^0, $(-7)^0$, -7^0.
2. Simplify, giving the answer with positive exponents: $\dfrac{x^{-2}y^5}{x^4y^{-3}}$.
3. Use the product law to explain why $10^{-3} \times 10^5 = 100$.

▼ **Answers**

1. $\mathbf{0.001}$; $\frac{1}{4}$; $\frac{1}{9}$ (even power: positive); $\mathbf{2^4 = 16}$; $\mathbf{1}$; $\mathbf{1}$; $\mathbf{-1}$ (the minus is outside the power).
2. $x^{-2-4}y^{5-(-3)} = x^{-6}y^8 = \dfrac{y^8}{x^6}$.
3. $10^{-3} \times 10^5 = 10^{-3+5} = 10^2 = 100$. In words: dividing by 1000 then multiplying by 100,000 is a net multiplication by 100.

3.4. Roots and fractional exponents

3.4.1. The problem

A square room has area 20 square metres. How long is each wall? We need to undo squaring. More generally, if we know x^n and want x, we need an inverse of raising to a power. Root notation does this; fractional exponents turn out to be the same idea in a form that obeys the exponent laws.

3.4.2. Intuition and vocabulary



Square root, cube root, nth root

A **square root** of a is a number whose square is a. Every positive number has two (3 and -3 are both square roots of 9). The symbol $\sqrt{a}$ — the **radical sign** over the **radicand** a — means the **principal** (non-negative) square root: $\sqrt{9} = 3$. A **cube root** $\sqrt[3]{a}$ satisfies $\left(\sqrt[3]{a}\right)^3 = a$; every real number has exactly one real cube root ($\sqrt[3]{-8} = -2$). In general $\sqrt[n]{a}$ is the **nth root**; the small n is the **index** of the root.

Negative numbers have no real square root, because any real number squared is non-negative (positive × positive and negative × negative are both positive). That gap is the one the complex numbers fill (Chapter 15).

Geometrically, $\sqrt{a}$ is the side of a square of area a. The picture below constructs a square of area exactly 2 on a unit grid, so its side has length exactly $\sqrt{2}$ — a length you can draw perfectly but (Section 1.1) never write as a fraction.

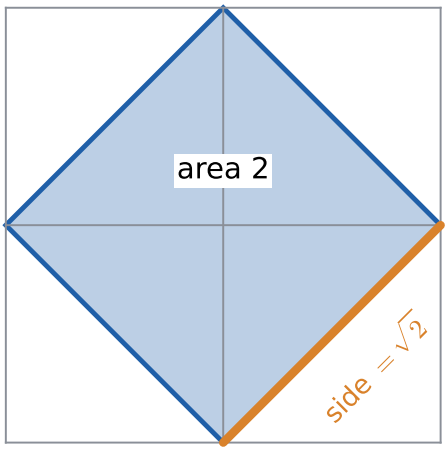


Figure 20. A square of area 2: it covers four half-squares. Its side is $\sqrt{2}$.

3.4.3. Why it works: fractional exponents

What should $9^{1/2}$ mean? Once again, demand that the laws keep working. Power-of-a-power says

$$\left(9^{1/2}\right)^2 = 9^{\frac{1}{2}\cdot 2} = 9^1 = 9.$$

So $9^{1/2}$ is a number whose square is 9: a square root of 9. We choose the positive one. In general, for $a > 0$,

$$a^{1/n} = \sqrt[n]{a}, \qquad a^{m/n} = \left(\sqrt[n]{a}\right)^m = \sqrt[n]{a^m}.$$

The second formula follows from $a^{m/n} = (a^{1/n})^m = (a^m)^{1/n}$, both of which are power-of-a-power. The denominator of the exponent is the root; the numerator is the power.

> *Rational exponents*
>
> For $a > 0$ and integers m, n with $n > 0$: $a^{1/n} = \sqrt[n]{a}$ and $a^{m/n} = (\sqrt[n]{a})^m$. All the laws of exponents continue to hold. So $\sqrt{x} = x^{1/2}$, $\dfrac{1}{\sqrt{x}} = x^{-1/2}$, $\sqrt[3]{x^2} = x^{2/3}$.
>
> Converting roots to fractional exponents is how calculus handles them (Volume 2, Chapter 17).

Squaring and square-rooting undo each other (for non-negative numbers), so their graphs are mirror images in the line $y = x$: a point (a, b) on one corresponds to (b, a) on the other. This mirror relationship between a function and its inverse is a central idea of Chapter 11.

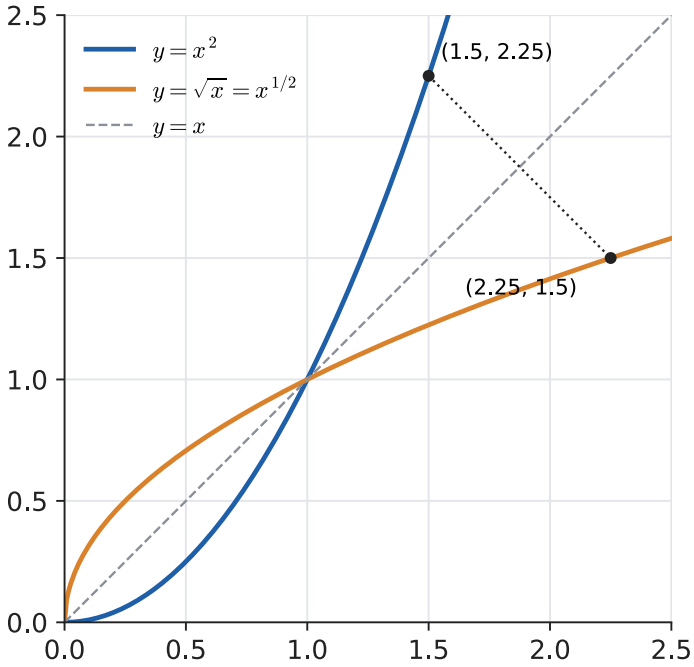


Figure 21. Squaring and square-rooting are inverse operations for $x \geq 0$: their graphs are reflections in $y = x$.

3.4.4. The method

Example 18. Procedure — evaluate $a^{m/n}$ by hand

Input: a positive base a and a fraction m/n (in lowest terms, $n > 0$).

Steps:

1. If $m < 0$, compute $a^{|m|/n}$ and take the reciprocal at the end.
2. Take the nth root first (it keeps numbers small): $r = \sqrt[n]{a}$.
3. Raise to the mth power: r^m.

Output: $a^{m/n}$.

3.4.5. Worked examples

$$8^{2/3} = \left(\sqrt[3]{8}\right)^2 = 2^2 = 4$$

$$16^{3/4} = \left(\sqrt[4]{16}\right)^3 = 2^3 = 8$$

$$27^{-1/3} = \frac{1}{27^{1/3}} = \frac{1}{3}$$

$$\left(\tfrac{4}{9}\right)^{3/2} = \left(\sqrt{\tfrac{4}{9}}\right)^3 = \left(\tfrac{2}{3}\right)^3 = \frac{8}{27}.$$

Root first versus power first: $16^{3/4}$ as $\sqrt[4]{16^3} = \sqrt[4]{4096}$ gives the same 8, but finding the fourth root of 4096 by hand is far harder. Root first is the right choice whenever the root comes out exactly.

3.4.6. Alternative approach: computing a root numerically

When the root is not exact ($\sqrt{20}$), you can estimate it. $\sqrt{20}$ lies between $\sqrt{16} = 4$ and $\sqrt{25} = 5$, closer to 4. A classical algorithm, the **Babylonian method**, improves any guess g by averaging it with $20/g$: if g is too big then $20/g$ is too small, and the average is better than either.

$$
\begin{aligned}
g_0 &= 4.5 \\
g_1 &= \tfrac{1}{2}\left(4.5 + \tfrac{20}{4.5}\right) = \tfrac{1}{2}(4.5 + 4.4444) = 4.47222 \\
g_2 &= \tfrac{1}{2}\left(4.47222 + \tfrac{20}{4.47222}\right) = \tfrac{1}{2}(4.47222 + 4.47205) = 4.47214
\end{aligned}
$$

Two steps already give $\sqrt{20} \approx 4.47214$ (true value $4.472135955\ldots$). The number of correct digits roughly doubles each step. This is Newton's method in disguise; Volume 2, Chapter 18 and Volume 2, Chapter 32 explain why it converges so fast.

3.4.7. Failure modes



Roots: the classic errors

- $\sqrt{a+b} \neq \sqrt{a} + \sqrt{b}$. Test: $\sqrt{9+16} = 5$, but $3 + 4 = 7$. (A root is a power, and powers do not distribute over sums.)
- $\sqrt{x^2} = |x|$, not x. If $x = -3$, $\sqrt{(-3)^2} = \sqrt{9} = 3$. The principal root is never negative (Section 1.7).
- "$x^2 = 9$ so $x = 3$" loses a solution: $x = \pm 3$ (read $\pm$ as "plus or minus"). Solving with a square root needs the $\pm$; the symbol $\sqrt{9}$ alone does not have it.
- Fractional powers of negative numbers are treacherous. $(-8)^{1/3}$ "should" be -2, but $(-8)^{2/6}$ would then be $\sqrt[6]{64} = 2$. The laws only hold for positive bases. Python's `(-8)**(1/3)` returns a complex number, `(1.0000000000000002+1.7320508075688772j)`; use `-(8**(1/3))` or `math.cbrt` (Python 3.11+) for the real cube root.

3.4.8. Where you'd use this

- **Everyday.** The square room with area 20 m² has walls of $\sqrt{20} \approx 4.47$ m. A TV's diagonal comes from Pythagoras (Chapter 9): a screen 30 by 40 inches is $\sqrt{30^2 + 40^2} = \sqrt{2500} = 50$ inches across.
- **Music.** Equal temperament splits an octave (a doubling of frequency) into 12 equal ratios, so each semitone multiplies the frequency by $2^{1/12} \approx 1.0595$. Concert A is 440 Hz; the next note up, A♯, is $440 \times 2^{1/12} \approx 466.2$ Hz.
- **Physics.** An object dropped from height h (ignoring air resistance) hits the ground after $t = \sqrt{2h/g}$: from 20 m, $\sqrt{40/9.81} \approx 2.0$ s. A pendulum of length 1 m has period $2\pi\sqrt{L/g} \approx 2.0$ s, which is why old clocks had metre-long pendulums.
- **Finance.** Fractional exponents turn total growth into a yearly rate. An investment that doubles in 10 years grew by a factor $2^{1/10} \approx 1.072$ each year: about 7.2% a year.

- **Computing.** The distance between two points in a game or a map is $\sqrt{\Delta x^2 + \Delta y^2}$. Code that only needs to know which of two objects is *closer* compares the squared distances instead, skipping the square root entirely. Root-mean-square error, the standard measure of a regression model's accuracy, is a square root of an average of squares (Volume 3, Chapter 31).

Computer-science connection: roots in code

`math.sqrt(x)` is a single hardware instruction on modern CPUs and is correctly rounded (the closest double to the true root). `x ** 0.5` usually gives the same result but goes through a general power routine. For integers, `math.isqrt(n)` returns $\lfloor \sqrt{n} \rfloor$ exactly (the brackets $\lfloor \rfloor$ mean "round down"), with no floating-point error — exactly what trial division (Section 1.8) needs. The famous "fast inverse square root" from *Quake III* computed $x^{-1/2}$ — needed to normalise vectors for lighting — with a bit-level hack on the float's exponent followed by one Newton step.



Where the radical sign came from

"Radical" comes from Latin *radix*, "root" (as in radish). The symbol √ is thought to derive from a stylised lowercase *r*, first printed in Christoff Rudolff's *Coss* of 1525. The bar over the radicand (the **vinculum**) was added by Descartes in 1637 to show how far the root extends — the same bar that became the fraction bar.



Root finder

Drag a square's side until its area matches a target (say 20) and watch the side length converge; then switch to the Babylonian method and step it, showing each guess and its partner $20/g$ closing in from opposite sides of the true root.

Try it: Babylonian square-root stepper

3.4.9. Exercises

1. Evaluate: $\sqrt{144}$, $\sqrt[3]{-125}$, $32^{1/5}$, $32^{3/5}$, $9^{-3/2}$.
2. Write as a power of x: $\sqrt{x}$, $\dfrac{1}{x\sqrt{x}}$, $\sqrt[4]{x^3}$.
3. Solve $x^2 = 50$ exactly and approximately (two Babylonian steps from $g_0 = 7$). *Try it:* Step √50 from 7 .

▼ **Answers**

1. 12; -5; 2; $2^3 = 8$; $\frac{1}{(\sqrt{9})^3} = \frac{1}{27}$.
2. $x^{1/2}$; $x^{-1} \cdot x^{-1/2} = x^{-3/2}$; $x^{3/4}$.
3. $x = \pm\sqrt{50} = \pm 5\sqrt{2}$ (see Section 3.5). $g_1 = \frac{1}{2}(7 + \frac{50}{7}) = \frac{1}{2}(7 + 7.142857) = 7.071429$; $g_2 = \frac{1}{2}(7.071429 + 7.070707) = 7.071068$. So $x \approx \pm 7.0711$.

3.5. Surds: simplifying radicals and rationalising denominators

3.5.1. The problem

Your calculator says $\sqrt{50} = 7.0710678\ldots$ and $\frac{1}{\sqrt{2}} = 0.7071067\ldots$. Notice anything? The digits are the same. Exact forms make such relationships visible — $\sqrt{50} = 5\sqrt{2}$ and $\frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2}$ — while decimals hide them. In geometry, trigonometry, and the quadratic formula, exact answers involving roots appear constantly, and you need to be able to put them in a standard simplest form.



Surd, radical form

A **surd** (UK usage) or **radical** (US usage) is an irrational root such as $\sqrt{2}$ or $\sqrt[3]{5}$, left in exact form rather than as a decimal. **Simplest radical form** means: no perfect-square factor under a square root, no root in a denominator, and no fraction under a root. **Like surds** have the same radicand ($3\sqrt{2}$ and $5\sqrt{2}$) and can be added like like terms.

3.5.2. Why it works

For $a, b \geq 0$, the power-of-a-product law with exponent $\frac{1}{2}$ gives

$$\sqrt{ab} = (ab)^{1/2} = a^{1/2}\,b^{1/2} = \sqrt{a}\,\sqrt{b}, \qquad \sqrt{\frac{a}{b}} = \frac{\sqrt{a}}{\sqrt{b}} \quad (b > 0).$$

So we can split a radicand into a perfect square times something else and pull the perfect square out.

3.5.3. The method

Example 19. Procedure — simplify a square root of a positive integer

Input: $\sqrt{n}$ for a positive integer n.

Steps:

1. Find the prime factorisation of n (Section 1.8).
2. Pair up equal primes. Each pair $p \cdot p = p^2$ comes out of the root as a single p.
3. Multiply the extracted primes together (outside) and the unpaired primes together (inside).

Output: $k\sqrt{m}$ where m has no square factor greater than 1.

Example 20. Procedure — rationalise a denominator

Input: a fraction with a square root in the denominator.

Steps:

1. If the denominator is $c\sqrt{b}$: multiply top and bottom by $\sqrt{b}$, using $\sqrt{b}\sqrt{b} = b$.
2. If the denominator is $a + \sqrt{b}$ (or $a - \sqrt{b}$): multiply top and bottom by the **conjugate** $a - \sqrt{b}$ (or $a + \sqrt{b}$), using $(a + \sqrt{b})(a - \sqrt{b}) = a^2 - b$.
3. Simplify.

Output: an equivalent fraction with a rational denominator.

The conjugate trick works because of the **difference of two squares**, $(x + y)(x - y) = x^2 - y^2$, which you can check with the distributive law: $x^2 - xy + yx - y^2 = x^2 - y^2$. The cross terms cancel, and squaring kills the root.

3.5.4. Worked examples

Simplify $\sqrt{72}$.

$$\begin{aligned} \sqrt{72} &= \sqrt{2 \cdot 2 \cdot 2 \cdot 3 \cdot 3} && \text{(factorise: } 72 = 2^3 \cdot 3^2\text{)} \\ &= \sqrt{(2 \cdot 2)(3 \cdot 3) \cdot 2} && \text{(pair up)} \\ &= 2 \cdot 3 \cdot \sqrt{2} = 6\sqrt{2}. \end{aligned}$$

The shortcut, when you can spot it, is to find the largest perfect-square factor: $72 = 36 \times 2$, so $\sqrt{72} = \sqrt{36}\sqrt{2} = 6\sqrt{2}$.

Add surds: $\sqrt{50} + \sqrt{18} - \sqrt{8} = 5\sqrt{2} + 3\sqrt{2} - 2\sqrt{2} = 6\sqrt{2}$. Only after simplifying are they visibly like surds.

Rationalise: $\dfrac{6}{\sqrt{3}}$.

$$\frac{6}{\sqrt{3}} = \frac{6}{\sqrt{3}} \cdot \frac{\sqrt{3}}{\sqrt{3}} = \frac{6\sqrt{3}}{3} = 2\sqrt{3}.$$

Rationalise with a conjugate: $\dfrac{4}{3 - \sqrt{5}}$.

$$\begin{aligned}\frac{4}{3-\sqrt{5}} &= \frac{4}{3-\sqrt{5}}\cdot\frac{3+\sqrt{5}}{3+\sqrt{5}} && \text{(multiply by 1, in the form conjugate/conjugate)}\\ &= \frac{4(3+\sqrt{5})}{3^2-(\sqrt{5})^2} && \text{(difference of two squares)}\\ &= \frac{4(3+\sqrt{5})}{9-5} = \frac{4(3+\sqrt{5})}{4}\\ &= 3+\sqrt{5}.\end{aligned}$$

3.5.5. Choosing a method, and why bother

- For simplifying: if you can see a perfect-square factor (4, 9, 16, 25, 36, 49, 64, 81, 100, ...), use it directly; otherwise factorise.
- Rationalising was originally about **hand computation**: $\frac{1}{\sqrt{2}} = \frac{1}{1.41421\ldots}$ is a horrible long division, while $\frac{\sqrt{2}}{2} = \frac{1.41421\ldots}{2}$ is easy. Today the reason is mainly **canonical form** (Section 2.2): it makes equal answers look equal, so you can compare your answer with someone else's. The conjugate technique itself remains important — the same trick evaluates limits in calculus (Volume 2, Chapter 16) and divides complex numbers (Chapter 15).

A surd you have handled today. A4 paper is 210 mm by 297 mm, and $297/210 \approx 1.414 \approx \sqrt{2}$. The ISO paper sizes are designed so that folding a sheet in half gives the next size down with exactly the same shape: if the sides are 1 and r, the half-sheet has sides $\frac{r}{2}$ and 1, and keeping the ratio means $r/1 = 1/(r/2)$, so $r^2 = 2$ and $r = \sqrt{2}$. That is why a photocopier can shrink A3 to A4 at "71%": $\frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2} \approx 0.707$.



Surd mistakes

$\sqrt{2}+\sqrt{3} \neq \sqrt{5}$ (unlike surds do not combine; $1.414 + 1.732 = 3.146 \neq 2.236$). $2\sqrt{3}$ means $2 \times \sqrt{3}$, not $\sqrt[2]{3}$ or $\sqrt{6}$. And $\sqrt{ab} = \sqrt{a}\sqrt{b}$ needs $a, b \geq 0$: "$1 = \sqrt{1} = \sqrt{(-1)(-1)} = \sqrt{-1}\sqrt{-1} = -1$" is a famous fake proof that breaks exactly there.

3.5.6. Exercises

1. Simplify: $\sqrt{48}$, $\sqrt{200}$, $3\sqrt{20} - \sqrt{45}$.
2. Rationalise: $\dfrac{10}{\sqrt{5}}$, $\dfrac{1}{\sqrt{2}+1}$.
3. A square has diagonal 10 cm. Find its side in simplest radical form. (Diagonal of a square with side s is $s\sqrt{2}$, by Pythagoras, Chapter 9.)

▼ **Answers**

1. $\sqrt{16\cdot 3} = 4\sqrt{3}$; $\sqrt{100\cdot 2} = 10\sqrt{2}$; $3\cdot 2\sqrt{5} - 3\sqrt{5} = 3\sqrt{5}$.
2. $\frac{10\sqrt{5}}{5} = 2\sqrt{5}$. $\frac{1}{\sqrt{2}+1}\cdot\frac{\sqrt{2}-1}{\sqrt{2}-1} = \frac{\sqrt{2}-1}{2-1} = \sqrt{2}-1$.

3. $s\sqrt{2} = 10$, so $s = \frac{10}{\sqrt{2}} = \frac{10\sqrt{2}}{2} = 5\sqrt{2}$ cm (about 7.07 cm).

US curriculum

Square roots and cube roots of perfect squares and cubes, and the irrationality of $\sqrt{2}$, are **grade 8** (8.EE.2, 8.NS). Simplifying radicals and rationalising denominators are **Algebra I** (often) and **Algebra II** (always); rational exponents are **Algebra II** (N-RN.1–2). In the UK, "surds" is a GCSE Higher and A-level topic.

3.6. Scientific notation and significant figures

3.6.1. The problem

The mass of the Earth is 5,970,000,000,000,000,000,000,000 kg. The mass of an electron is 0.000 000 000 000 000 000 000 000 000 911 kg. Written out, these are unreadable, impossible to compare at a glance, and easy to miscount. Worse, they claim false precision: is the Earth's mass really known to the last kilogram?

3.6.2. Intuition

Separate a number into **how big** (a power of ten) and **which digits** (a number between 1 and 10). The Earth is 5.97×10^{24} kg: "about 6, followed by 24 places". The electron is 9.11×10^{-31} kg. The exponent carries the size; the leading number carries the precision.



Scientific notation

A number is in **scientific notation** (UK: **standard form**) when written as $m \times 10^{k}$ with $1 \le |m| < 10$ and k an integer. m is the **significand** (or **mantissa** or **coefficient**). Programming languages and calculators write 5.97×10^{24} as `5.97e24` (**E notation**). **Engineering notation** restricts k to multiples of 3, matching the prefixes kilo, mega, milli, micro.

3.6.3. The method

Example 21. Procedure — convert to scientific notation

Input: a non-zero decimal number.

Steps:

1. Move the decimal point so that exactly one non-zero digit is to its left. This is m.
2. Count how many places the point moved: k. If you moved it left (the number was large), k is positive; if right (the number was small), k is negative.
3. Write $m \times 10^{k}$, keeping only the significant digits in m.

Output: $m \times 10^{k}$.

Examples. $384\,400$ km: point moves 5 places left, 3.844×10^{5} km. $0.000\,72$ s: point moves 4 places right, 7.2×10^{-4} s.

Arithmetic. Multiply and divide the significands, and combine the powers of ten with the exponent laws; then re-normalise.

$$(6 \times 10^{4})(5 \times 10^{6}) = (6 \times 5) \times 10^{4+6} = 30 \times 10^{10} = 3.0 \times 10^{11}$$

$$\frac{4.8 \times 10^{7}}{1.6 \times 10^{-2}} = \frac{4.8}{1.6} \times 10^{7-(-2)} = 3.0 \times 10^{9}$$

$$\frac{5.97 \times 10^{24}}{9.11 \times 10^{-31}} = 0.655 \times 10^{55} = 6.55 \times 10^{54}.$$

The last line says the Earth outweighs an electron by a factor of about 6.6×10^{54} — a comparison that would be hopeless in long form. Adding and subtracting require matching exponents first: $3.2 \times 10^{5} + 4 \times 10^{3} = 3.2 \times 10^{5} + 0.04 \times 10^{5} = 3.24 \times 10^{5}$.

3.6.4. Significant figures



Significant figures

The **significant figures** (s.f., or **significant digits**) of a measured number are the digits that carry real information about its precision. Rules: non-zero digits are significant; zeros *between* non-zero digits are significant (4007: 4 s.f.); leading zeros are not (0.0072: 2 s.f.); trailing zeros after a decimal point are (2.50: 3 s.f.); trailing zeros in a whole number are ambiguous (1500: 2, 3, or 4?) — scientific notation removes the ambiguity: 1.5×10^{3} vs 1.500×10^{3}.

The working rule for calculations: **a product or quotient should be reported to the fewest significant figures of its inputs.** If a car travels 125.0 km in 1.5 h, the speed is $125.0/1.5 = 83.333\ldots$, reported as 83 km/h (2 s.f., limited by 1.5). For sums and differences, the limiting factor is instead the fewest *decimal places*. These are rules of thumb; proper error propagation is part of Volume 3, Chapter 31 and Volume 2, Chapter 32 .



False precision

A calculator showing 83.33333333 has not measured the speed to ten digits. Reporting all of them misleads the reader about how well you know the answer. The converse also bites: dropping meaningful trailing zeros (writing 2.5 for a measurement known to be 2.50) throws away information.

3.6.5. Orders of magnitude

The exponent k in $m \times 10^{k}$ (rounded according to whether m is nearer 1 or 10) is the **order of magnitude**. Two quantities "differ by three orders of magnitude" when one is about 1000 times the other. Thinking in orders of magnitude is how physicists and engineers make sense of scales from the subatomic to the cosmic — and how you do the Fermi estimates of Section 1.10.

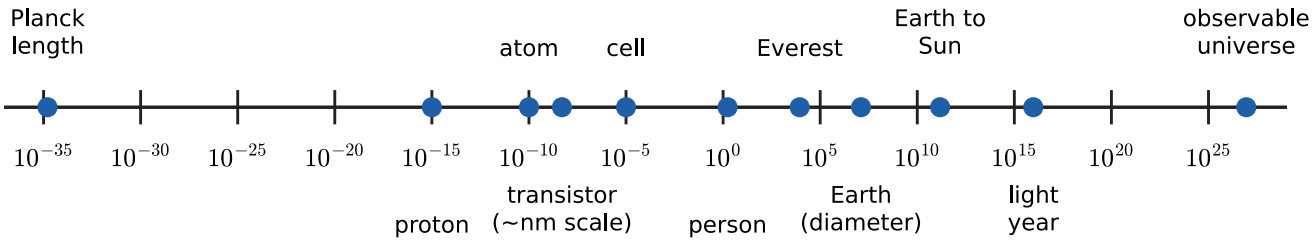


Figure 22. Lengths in the universe span over 60 orders of magnitude. On this scale, each tick is a factor of 10^5.

A scale like this, in which equal distances mean equal *ratios* rather than equal *differences*, is a **logarithmic scale**. You will meet it formally in Chapter 12; it is the natural scale for anything spanning many powers of ten (earthquakes, sound levels, pH, algorithm running times).

Computer-science connection: floats are scientific notation in binary

A `double` is stored as $\pm m \times 2^{e}$ with $1 \le m < 2$ — scientific notation in base 2 (Section 2.8). The 52 stored bits of m give about 15–16 significant decimal digits; the 11-bit exponent allows e from about -1022 to 1023, so doubles range from about 2.2×10^{-308} to 1.8×10^{308}. The consequences follow directly: adding numbers of very different magnitude loses the smaller one entirely (`1e16 + 1 == 1e16`), because the significand cannot hold digits 16 orders of magnitude apart. This is significant-figure reasoning, enforced by hardware.



Powers of Ten

Charles and Ray Eames's 1977 short film *Powers of Ten* starts with a picnic in Chicago and zooms out by a factor of 10 every ten seconds to 10^{24} m, then zooms in to a proton. It remains one of the best nine minutes ever made on scale. The SI prefixes recently had to be extended for the same reason: in 2022 **ronna** (10^{27}) and **quetta** (10^{30}) were added, partly because global data storage is heading toward those scales.



Scale zoomer

An infinite zoom from 10^{-35} m to 10^{27} m, with landmarks appearing as you pass their order of magnitude. A side panel shows the current scale in scientific notation, E notation, and the matching SI prefix, and lets you pin two objects to compare their ratio.

3.6.6. Exercises

1. Write in scientific notation: 93,000,000; 0.000 040 5; 6.02 billion.

2. Compute, giving answers in scientific notation: $(2.5 \times 10^{-3})(4 \times 10^{8})$; $\dfrac{9 \times 10^{5}}{3 \times 10^{-4}}$; $7.1 \times 10^{4} - 9 \times 10^{3}$.
3. How many significant figures: 0.00306, 3.060, 3.06×10^{-3}?
4. Light travels at 3.00×10^{8} m/s. The Sun is 1.50×10^{11} m away. How long does sunlight take to reach us?

▼ **Answers**

1. 9.3×10^{7}; 4.05×10^{-5}; 6.02×10^{9}.
2. $10 \times 10^{5} = 1.0 \times 10^{6}$; 3×10^{9}; $7.1 \times 10^{4} - 0.9 \times 10^{4} = 6.2 \times 10^{4}$.
3. 3; 4; 3.
4. $\frac{1.50 \times 10^{11}}{3.00 \times 10^{8}} = 0.500 \times 10^{3} = 5.00 \times 10^{2}$ s, which is 500 s, or about 8.3 minutes.

> **US curriculum**
>
> Scientific notation, including operations with it: **grade 8** (8.EE.3–4). Significant figures are taught mainly in high-school **chemistry** and **physics**. Logarithmic scales appear in **Algebra II** with logarithms.

3.7. Powers of two: the computer's native scale

3.7.1. The problem

Why is a "32-bit" program limited to 4 GB of memory? Why does a "2 TB" drive show up as 1.82 TB on your computer? Why are buffer sizes, hash table capacities, and texture dimensions so often 256, 1024, 4096? All of it follows from one fact: with n bits you can write exactly 2^{n} different patterns.

3.7.2. Why it works

Each bit has 2 possible values. Two bits give $2 \times 2 = 4$ patterns (00, 01, 10, 11); three bits give $2 \times 2 \times 2 = 8$; n bits give 2^{n}. (This "multiply the number of choices" rule is the multiplication principle of Volume 3, Chapter 24 .) So:

n	2^n	significance
8	256	values in a byte
10	1024	$\approx 10^3$ (kibi)
16	65 536	port numbers; old 16-bit address spaces
20	1 048 576	$\approx 10^6$ (mebi)
30	1 073 741 824	$\approx 10^9$ (gibi)
32	4 294 967 296	IPv4 addresses; 32-bit address space (4 GiB)
63	$\approx 9.22 \times 10^{18}$	largest signed 64-bit integer + 1
64	$\approx 1.84 \times 10^{19}$	64-bit patterns
128	$\approx 3.40 \times 10^{38}$	IPv6 addresses; UUIDs

A 32-bit pointer can name 2^{32} different bytes, which is $2^{32} = 2^2 \cdot 2^{30} = 4$ GiB: the 4 GB limit. An n-bit signed integer ranges from -2^{n-1} to $2^{n-1} - 1$ (Section 1.9). *Try it:* What 32 bits can address .

3.7.3. The approximation that makes it all mental

$2^{10} = 1024 \approx 10^3$. With the power-of-a-power law, every power of 2 can be estimated in your head:

$$2^n = 2^{10q+r} = (2^{10})^q \cdot 2^r \approx 10^{3q} \cdot 2^r$$

where q and r are the quotient and remainder of $n \div 10$. So $2^{32} = (2^{10})^3 \cdot 2^2 \approx 10^9 \times 4 = 4$ billion (true: 4.29 billion), and $2^{64} = (2^{10})^6 \cdot 2^4 \approx 10^{18} \times 16 = 1.6 \times 10^{19}$ (true: 1.84×10^{19}). The error grows with each factor of 2^{10}, since each contributes 2.4%.

3.7.4. KB versus KiB

That 2.4% is the source of a long-running confusion. Should a "kilobyte" be 1000 bytes (as "kilo" means everywhere else) or 1024 bytes (a convenient power of 2)? Both usages were common, so in 1998 the IEC introduced separate binary prefixes:



Decimal (SI) and binary (IEC) prefixes

SI (decimal): kilo (k) $= 10^3$, mega (M) $= 10^6$, giga (G) $= 10^9$, tera (T) $= 10^{12}$, peta (P) $= 10^{15}$. **IEC (binary):** kibi (Ki) $= 2^{10}$, mebi (Mi) $= 2^{20}$, gibi (Gi) $= 2^{30}$, tebi (Ti) $= 2^{40}$, pebi (Pi) $= 2^{50}$. So 1 KB $= 1000$ bytes and 1 KiB $= 1024$ bytes. Lower-case b means bits, upper-case B bytes: 1 Gbit/s is 125 MB/s.

Storage manufacturers use SI prefixes; Windows reports sizes in binary units but labels them "GB" and "TB"; macOS switched to SI units in 2009; RAM is always sold in binary sizes. The discrepancy grows with each prefix:

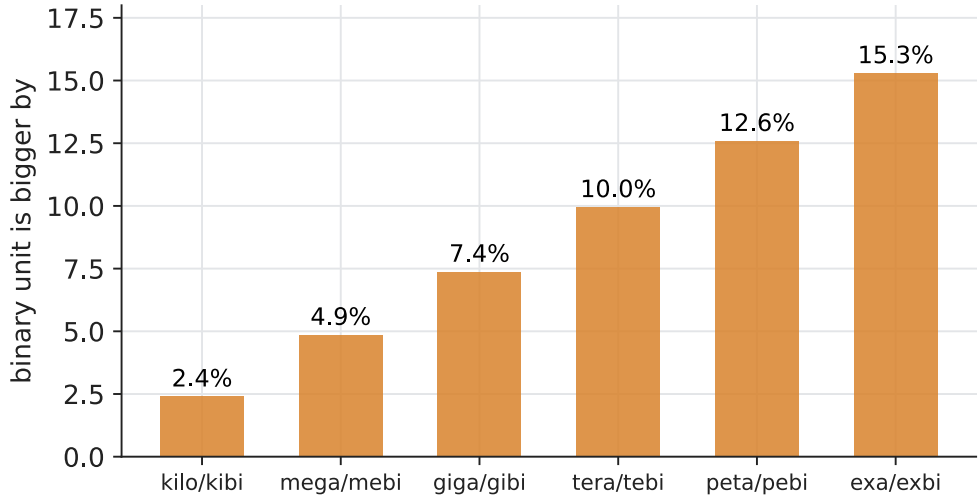


Figure 23. How much larger each binary unit is than its decimal namesake. The gap compounds: $(1.024)^k - 1$.

Worked example: the "missing" space. A drive sold as 2 TB holds 2×10^{12} bytes. In TiB: *Try it:* Convert a 2 TB drive .

$$\frac{2 \times 10^{12}\ \text{B}}{2^{40}\ \text{B/TiB}} = \frac{2 \times 10^{12}}{1.0995 \times 10^{12}} \approx 1.82\ \text{TiB}.$$

Nothing is missing; the same quantity is being reported in a 10% larger unit.

```python
def human_bytes(n, binary=True):
    """Format a byte count with SI (kB, MB, ...) or IEC (KiB, MiB, ...) prefixes."""
    base = 1024 if binary else 1000
    units = ["KiB", "MiB", "GiB", "TiB", "PiB"] if binary else ["kB", "MB", "GB", "TB", "PB"]
    if n < base:
        return f"{n} B"
    value, unit = float(n), "B"
    for u in units:
        if value < base:
            break
        value /= base
        unit = u
    return f"{value:.2f} {unit}"

disk = 2 * 10**12                       # a "2 TB" drive as sold
print(human_bytes(disk, binary=False))
print(human_bytes(disk, binary=True))
print(human_bytes(1536), human_bytes(1536, binary=False))
```

```
2.00 TB
1.82 TiB
1.50 KiB 1.54 kB
```

Computer-science connection: why powers of two everywhere

- **Addressing.** Memory chips are addressed by binary numbers, so capacities come naturally in powers of 2.
- **Fast arithmetic.** Multiplying or dividing by 2^k is a shift by k bits (`x << k`, `x >> k`), and `x % 2**k` is a mask (`x & (2**k - 1)`). Hash tables with power-of-two capacity can compute a bucket index with a single AND instead of a division.
- **Doubling growth.** Dynamic arrays (Python lists, `std::vector`, Java `ArrayList`) grow by a constant factor (often 1.5–2) when full. Because the sizes form a geometric sequence, the total copying cost over n appends is proportional to n — **amortised** constant time per append. Chapter 14 proves it with a geometric series.
- **Is it a power of two?** `n > 0 and n & (n - 1) == 0`: subtracting 1 from a power of two flips its single 1 bit to 0 and all lower bits to 1, so the AND is zero.



Rice on a chessboard

A legend tells of a sage who asked a king for one grain of rice on the first square of a chessboard, two on the second, four on the third, doubling each time. The last square alone needs $2^{63} \approx 9.2 \times 10^{18}$ grains; the whole board needs $2^{64} - 1 \approx 1.8 \times 10^{19}$ — roughly a thousand years of current world rice production. Note that $2^{64} - 1$ is also the largest unsigned 64-bit integer: the chessboard is a 64-bit register.

Try it: The 64-bit chessboard .



Bits and bytes calculator

A slider for bit width n from 1 to 128 that shows the number of patterns 2^n, the signed and unsigned ranges, the memory addressable with n-bit pointers, and a familiar real-world comparison ("more than the number of grains of sand on Earth"). A second panel converts any size among B, KB, KiB, MB, MiB, ..., highlighting the percentage gap.

Try it: Open the Bits and bytes calculator

3.7.5. Exercises

1. Without a calculator, estimate 2^{24} and 2^{50}.
2. How many bits are needed to give a unique ID to each of 10 billion devices?
3. A 16 GiB RAM module holds how many bytes, exactly, as a power of 2? About how many in GB?
4. Explain why `x & 7` gives the same result as `x % 8` for non-negative integers `x`.

▼ **Answers**

1. $2^{24} = 2^{20} \cdot 2^4 \approx 10^6 \times 16 = 1.6 \times 10^7$ (true: 16,777,216 — the number of 24-bit colours). $2^{50} = (2^{10})^5 \approx 10^{15}$ (true: 1.13×10^{15}).
2. Need $2^n \geq 10^{10}$. $2^{30} \approx 10^9$ is too small; $2^{33} \approx 8.6 \times 10^9$ is too small; $2^{34} \approx 1.7 \times 10^{10}$ suffices: 34 bits.
3. $16 \times 2^{30} = 2^4 \cdot 2^{30} = 2^{34} = 17\,179\,869\,184$ bytes ≈ 17.2 GB.
4. $7 = 111_2$. The AND keeps the lowest three bits of `x`, which (by place value, Section 1.3) represent exactly the remainder of `x` on division by $2^3 = 8$; all higher bits are multiples of 8.

3.8. A first look at exponential growth and logarithms

3.8.1. The problem

An algorithm that takes n^2 steps on an input of size n and another that takes 2^n steps both "get slower as n grows". Is the difference important? For $n = 10$, $n^2 = 100$ and $2^{10} = 1024$ — not very different. For $n = 100$, $n^2 = 10\,000$ while $2^{100} \approx 1.27 \times 10^{30}$. If each step took a nanosecond, the first finishes in 10 microseconds; the second would take about 4×10^{13} years, thousands of times the age of the universe.

3.8.2. Intuition

In n^2 the variable is the **base** and the exponent is fixed: a **power function** (or **polynomial**). In 2^n the variable is the **exponent** and the base is fixed: an **exponential function**. Exponential growth means a fixed **percentage** increase per step (here, 100% per step — doubling), and as we saw with compound interest (Section 2.5), repeated percentage increases snowball.

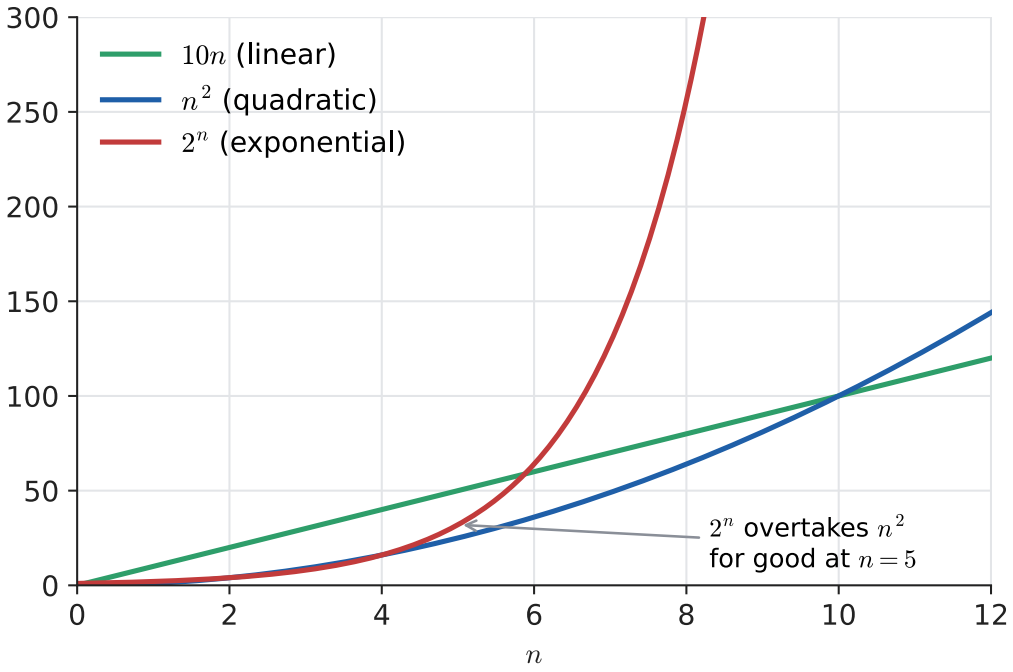


Figure 24. Exponential growth eventually overtakes any power. Here 2^n passes n^2 for good at $n = 5$ and leaves it far behind.



Exponential versus polynomial

An **exponential** expression has the variable in the exponent: 2^n, 1.05^t, 10^x. A **polynomial** (or **power**) expression has the variable in the base: n^2, n^3, x^{10}. For any base $b > 1$ and any power k, b^n eventually exceeds n^k and the ratio grows without bound. Exponential growth with base $b > 1$ has a **doubling time**; exponential decay (base between 0 and 1) has a **half-life**.

3.8.3. Running it backwards: the logarithm

Exponentiation answers "2 to the 10 is what?" (1024). The reverse question is "2 to the *what* is 1024?" The answer, 10, is the **logarithm base 2** of 1024:

$$\log_2 1024 = 10 \quad \text{because} \quad 2^{10} = 1024.$$

In general $\log_b x$ is the exponent you must put on b to get x. Logarithms turn multiplication into addition (since exponents add when powers multiply) and compress huge ranges into small ones. They deserve — and get — a full chapter, Chapter 12. Here are the two facts every programmer uses daily:

- **Bits needed.** To give distinct codes to N things you need $\lceil \log_2 N \rceil$ bits ($\lceil\ \rceil$ means "round up"). For 10 billion devices: $\log_2 10^{10} \approx 33.2$, so 34 bits — matching the exercise in Section 3.7.
- **Halving steps.** Binary search on N sorted items halves the range each step, so it needs about $\log_2 N$ steps: 20 steps for a million items, 30 for a billion. That $\log_2 N$ grows so slowly is the mirror image of 2^n growing so fast.

3.8.4. Where you'd use this

- **Savings and debt.** £1000 at 5% a year, compounded for 30 years, becomes $1000 \times 1.05^{30} \approx £4322$. The same arithmetic applied to an unpaid credit-card balance is why debt "suddenly" becomes unmanageable.
- **Biology.** Under ideal conditions *E. coli* divides every 20 minutes. One cell becomes $2^{30} \approx 1.07 \times 10^9$ cells after 30 doublings, i.e. just 10 hours. Food-safety rules about not leaving food at room temperature come from this calculation; the same doubling model is the starting point for epidemic forecasting.
- **Security.** A password of 8 lowercase letters has $26^8 \approx 2.1 \times 10^{11}$ possibilities. Each extra character multiplies the attacker's work by 26, while each extra character only adds one to your typing. Cryptography is built on exactly this asymmetry: the defender's cost grows linearly, the attacker's exponentially.

Computer-science connection: big-O teaser

Algorithm analysis classifies running times by growth rate: $O(\log n)$ (binary search), $O(n)$ (a single pass), $O(n \log n)$ (good sorting), $O(n^2)$ (nested loops), $O(2^n)$ (trying every subset). The notation $O(\cdot)$, read "big-O", means "grows no faster than, up to a constant factor"; it is defined precisely in Volume 2, Chapter 16 . The practical boundary is between **polynomial** and **exponential**: polynomial algorithms usually scale to large inputs; exponential ones hit a wall around $n = 40$ to 60 no matter how fast the hardware. Doubling your computer's speed lets an $O(2^n)$ algorithm handle inputs just **one** larger.



Exponential intuition fails everyone

A lily pad patch doubles in size every day and covers the pond on day 30. On what day was it half covered? Day 29 — not day 15. Exponential processes look negligible for most of their history and then appear to explode. That is why early-stage epidemics, compound debts, and viral software growth are so routinely underestimated: on a linear chart, nearly all the change happens at the end.



Growth race

Choose growth functions (n, $n \log n$, n^2, n^3, 2^n, $n!$) and a "computer speed" in steps per second. A race shows the largest n each can handle in one second, one hour, and one year. Toggle the vertical axis between linear and logarithmic and watch the exponential curve become a straight line.

Try it: Growth curves on log axis · Growth race: one second · Growth race: one year

3.8.5. Exercises

1. Compute $\log_2 64$, $\log_{10} 1000$, $\log_3 81$, $\log_2 \frac{1}{8}$.

2. About how many steps does binary search need for 10^{12} items?
3. Which is larger for $n = 30$: n^5 or 2^n? For $n = 20$?

▼ **Answers**

1. 6; 3; 4; -3 (since $2^{-3} = \frac{1}{8}$).
2. $10^{12} = (10^3)^4 \approx (2^{10})^4 = 2^{40}$: about 40 steps.
3. $n = 30$: $30^5 = 24\,300\,000 \approx 2.4 \times 10^7$ and $2^{30} \approx 1.07 \times 10^9$: the exponential is larger.
   $n = 20$: $20^5 = 3\,200\,000$ and $2^{20} \approx 1\,048\,576$: the power is still larger. (They cross near $n \approx 23$.)

3.9. Exponentiation by squaring

3.9.1. The problem

To compute 3^{13} the obvious way, start with 3 and multiply by 3 twelve more times: 12 multiplications. To compute a^n that way takes $n - 1$ multiplications. In cryptography (RSA, Diffie–Hellman) exponents have hundreds of digits — $n \approx 2^{2048}$. At a billion multiplications per second the obvious method would not finish before the stars burn out. Yet your browser does it many times a second. How?

3.9.2. Intuition

Squaring doubles the exponent: $3 \to 3^2 \to 3^4 \to 3^8 \to 3^{16}$ takes only four multiplications to reach the 16th power. Any exponent can be assembled from such powers, because any number is a sum of powers of two — that is exactly what its binary representation says (Section 1.3):

$$13 = 8 + 4 + 1 = 1101_2, \qquad \text{so} \qquad 3^{13} = 3^{8+4+1} = 3^8 \cdot 3^4 \cdot 3^1.$$

The step from sum to product is the product law of exponents, read backwards. Compute the repeated squares $3^1, 3^2, 3^4, 3^8$, and multiply together those whose bit is 1.

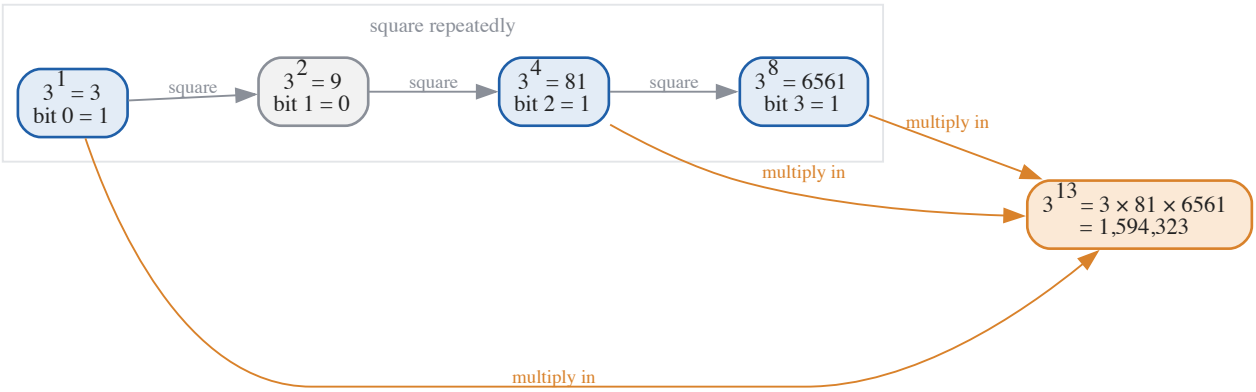


Figure 25. Computing 3^{13}: square repeatedly, then multiply together the powers whose binary digit of 13 is 1.

3.9.3. The method

Example 22. Procedure — exponentiation by squaring (right-to-left binary method)

Input: a base a and an integer exponent $n \geq 0$.

Steps:

1. Set $\mathbf{result} = 1$ and $\mathbf{square} = a$.
2. While $n > 0$:
   a. If n is odd (its lowest binary digit is 1), set $\mathbf{result} = \mathbf{result} \times \mathbf{square}$.
   b. Set $n = \lfloor n/2 \rfloor$ (drop the lowest bit; $\lfloor\ \rfloor$ means round down).
   c. If $n > 0$, set $\mathbf{square} = \mathbf{square} \times \mathbf{square}$.

Output: $\mathbf{result} = a^n$.

Try it: 3^{13} by repeated squaring .

3.9.4. Worked example: 3^{13}

Each row is one pass through the loop. "Bit" is the lowest binary digit of n at the start of the pass.

n	bit	square ($= 3^{2^k}$)	result after this pass	action
13	1	3	$1 \times 3 = 3$	multiply in, then square
6	0	9	3	skip, then square
3	1	81	$3 \times 81 = 243$	multiply in, then square
1	1	6561	$243 \times 6561 = 1\,594\,323$	multiply in; n becomes 0, stop

So $3^{13} = 1\,594\,323$, using 3 squarings and 3 multiplications into the result: 6 instead of 12. (Counting the first "1×3", which a careful implementation can skip.)

Why it is correct. At every stage, the invariant $\text{result} \times \text{square}^{\,n} = a^N$ holds, where N is the original exponent. Initially it is $1 \times a^N$. If n is even, squaring the square and halving n leaves square^n unchanged (power-of-a-power law: $(s^2)^{n/2} = s^n$). If n is odd, we first move one factor of square into the result, making the remaining exponent even. When n reaches 0, $\text{square}^0 = 1$ and the result is a^N. (Loop invariants are a proof technique; Volume 3, Chapter 22 develops them.)

Cost. The loop runs once per binary digit of n, about $\log_2 n$ times, with at most two multiplications each: at most about $2 \log_2 n$ multiplications instead of $n - 1$. For a 2048-bit exponent, that is about 4096 multiplications instead of 2^{2048}.

3.9.5. Implementation

```python
def fast_pow(a, n):
    """Compute a**n for an integer n >= 0 by repeated squaring.

    Returns (result, number_of_multiplications) so we can see the saving.
    """
    result, square, mults = 1, a, 0
    while n > 0:
        if n & 1:                # lowest bit of n is 1: this power is needed
            result *= square
            mults += 1
        n >>= 1                  # move on to the next bit
        if n:                    # square only if more bits remain
            square *= square
            mults += 1
    return result, mults

print(fast_pow(3, 13))
print(fast_pow(2, 64)[0] == 2**64, fast_pow(2, 64)[1])
print(fast_pow(7, 0), fast_pow(1.0001, 10_000)[0])
```

```
(1594323, 6)
True 7
(1, 0) 2.718145926824356
```

($1.0001^{10\,000}$ being close to $e = 2.71828\ldots$ is no coincidence; Chapter 12 explains it.)

In cryptography the numbers would become astronomically large, so every multiplication is followed by taking the remainder modulo some m, which keeps all intermediate values below m (why this is legitimate is Volume 3, Chapter 23 's business):

```python
def mod_pow(a, n, m):
    """Compute a**n mod m by repeated squaring, keeping every number below m."""
    result, square = 1, a % m
    while n > 0:
        if n & 1:
            result = result * square % m
        square = square * square % m
        n >>= 1
    return result

print(mod_pow(3, 13, 1000), 3**13 % 1000)
print(mod_pow(2, 10**18, 10**9 + 7) == pow(2, 10**18, 10**9 + 7))
```

```
323 323
True
```

Python's built-in three-argument `pow(a, n, m)` is exactly this algorithm. Note that `mod_pow(2, 10**18, ...)` takes about 60 loop iterations; the naive method would take 10^{18}.

3.9.6. Alternative approaches and choosing

- **Naive repeated multiplication:** simplest, fine for small exponents (n up to a few dozen), and has the virtue of being obviously correct.
- **Recursive halving:** $a^n = (a^{n/2})^2$ if n is even, $a \cdot a^{n-1}$ if odd. The same cost, and the clearest way to *explain* the idea; but recursion depth grows with $\log_2 n$.
- **Left-to-right binary:** scan the bits from the most significant end, squaring the result each step and multiplying by a when the bit is 1. Same count; preferred in hardware because it always multiplies by the same fixed a.
- **Floating-point `a ** n` or `math.pow`:** for real (non-integer) results, the library computes $e^{n \ln a}$ using logarithms (Chapter 12), which is constant time but only approximately correct. For exact integer results, use integer exponentiation.

Choose integer squaring (or `pow(a, n, m)`) whenever exactness matters; choose floating-point power when you need real-valued results and can tolerate rounding.



> *Overflow and timing*
>
> In languages with fixed-width integers, `square *= square` overflows quickly — 3^{40} does not fit in 64 bits — and silently wraps (Section 1.9). Reduce modulo m at every step, or use a big-integer type. In cryptographic code there is a subtler failure: because the method does extra work for each 1 bit, the running time leaks the number of 1 bits in a secret exponent. Real cryptographic libraries use **constant-time** variants (such as the Montgomery ladder) that do the same work for every bit.

Computer-science connection: squaring beyond numbers

The method works for any associative multiplication, not only numbers. Raising a 2×2 matrix to the nth power by squaring computes the nth Fibonacci number in about $\log_2 n$ matrix multiplications (Volume 2, Chapter 27). Squaring a graph's adjacency matrix counts paths of doubling length (Volume 3, Chapter 25). Even string repetition `s * n` can be implemented by doubling. Associativity — the dull-looking law from Section 1.4 — is exactly what makes regrouping $a \cdot a \cdots a$ into squares legitimate.



An ancient algorithm

The binary method of exponentiation appears in Pingala's *Chandaḥśāstra*, an Indian text on poetic metre from around the 2nd century BCE, where it is used to count metrical patterns. The closely related "Russian peasant" or ancient Egyptian method of multiplication — doubling one number and halving the other — is the same idea with addition in place of multiplication: $13 \times 3 = 3 \cdot 8 + 3 \cdot 4 + 3 \cdot 1$.



Square-and-multiply stepper

Enter a base and exponent. The exponent's bits are shown as a row of switches; stepping through the loop lights up the current bit, shows the square and the result, and counts multiplications against the naive method. A slider pushes the exponent to thousands of bits (with a modulus) to show the naive count becoming astronomically large while the fast count grows by one or two per bit.

Try it: Open the square-and-multiply stepper · Huge powers mod m, instantly

3.9.7. Exercises

1. Using the binary method, compute 2^{21} by hand, showing the table of n, bit, square, and result. *Try it:* Check 2^{21} step by step .
2. How many multiplications does the method use (at most) for $n = 1000$? How many does the naive method use?
3. Compute $7^{10} \bmod 13$ with the binary method, reducing modulo 13 at every step. ($\bmod$ means "remainder on division by".) *Try it:* Check $7^{10} \bmod 13$.

▼ **Answers**

1. $21 = 10101_2$.

n	bit	square	result
21	1	2	2
10	0	4	2
5	1	16	32
2	0	256	32
1	1	65536	$32 \times 65536 = 2\,097\,152$

So $2^{21} = 2\,097\,152$.

2. $1000 = 1111101000_2$ has 10 bits, six of them 1: 9 squarings and 6 multiplications into the result, 15 in all (or 14 if the first $1 \times a$ is skipped), versus 999 naively.
3. $10 = 1010_2$. Squares mod 13: 7; $7^2 = 49 \equiv 10$ (since $49 = 3 \cdot 13 + 10$); $10^2 = 100 \equiv 9$; $9^2 = 81 \equiv 3$. Bits (from the right) 0, 1, 0, 1 select the 2nd and 4th squares: $10 \times 3 = 30 \equiv 4$. So $7^{10} \bmod 13 = 4$. (The symbol $\equiv$ means "leaves the same remainder"; see Volume 3, Chapter 23.)

US curriculum

Exponential growth and decay: **Algebra I** (F-LE). Logarithms: **Algebra II** and **Precalculus**. Exponentiation by squaring and big-O notation are not school mathematics; they belong to **AP Computer Science** (informally) and college **algorithms** and **discrete mathematics** courses.

3.10. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Root finding, step by step
- Programmer's calculator
- Big-O explorer and benchmark
- Number theory step by step
- Number theory

3.11. Mastery check

1. Simplify, giving positive exponents: $\dfrac{(2x^{3}y^{-1})^{3}}{4x^{-2}y^{2}}$.
2. Explain, using the quotient law, why $5^{0}=1$, and evaluate $\left(\frac{3}{4}\right)^{-2}$.
3. Evaluate $64^{2/3}$ and $81^{-3/4}$ without a calculator.
4. Simplify $\sqrt{75}+\sqrt{12}$ and rationalise $\dfrac{2}{\sqrt{7}-\sqrt{5}}$.
5. Compute $\dfrac{(3.0\times10^{5})(8.0\times10^{-2})}{1.2\times10^{-4}}$ in scientific notation, to 2 significant figures.
6. A cloud provider bills 500 GiB. How many GB is that, to 3 s.f.? How many bits does a 32-bit address bus address, and how many GiB is that in bytes?
7. Which grows faster as n gets large, n^{10} or 1.1^{n}? About how many steps does binary search take on 4 billion items?
8. Compute 5^{11} by exponentiation by squaring, showing the table, and say how many multiplications you used.

Decision rule: All correct without hesitation → you have finished the Foundations part; move on to Chapter 4. Stuck on 1–2 → reread the matching section (Section 3.2, Section 3.3, Section 3.4, Section 3.5, Section 3.6, Section 3.7, Section 3.8, Section 3.9). Stuck on more → work the section exercises first; exponent fluency is assumed constantly from here on.

▼ **Mastery check answers**

1. Numerator $8x^{9}y^{-3}$. Divide: $\frac{8}{4}x^{9-(-2)}y^{-3-2}=2x^{11}y^{-5}=\dfrac{2x^{11}}{y^{5}}$.
2. $1=\frac{5^{n}}{5^{n}}=5^{n-n}=5^{0}$. $\left(\frac{3}{4}\right)^{-2}=\left(\frac{4}{3}\right)^{2}=\frac{16}{9}$.
3. $(\sqrt[3]{64})^{2}=4^{2}=16$. $\frac{1}{(\sqrt[4]{81})^{3}}=\frac{1}{3^{3}}=\frac{1}{27}$.
4. $5\sqrt{3}+2\sqrt{3}=7\sqrt{3}$. $\frac{2}{\sqrt{7}-\sqrt{5}}\cdot\frac{\sqrt{7}+\sqrt{5}}{\sqrt{7}+\sqrt{5}}=\frac{2(\sqrt{7}+\sqrt{5})}{7-5}=\sqrt{7}+\sqrt{5}$.
5. Numerator: $24\times10^{3}=2.4\times10^{4}$. Divide: $\frac{2.4}{1.2}\times10^{4-(-4)}=2.0\times10^{8}$.
6. $500\times2^{30}=536\,870\,912\,000$ bytes ≈ 537 GB. A 32-bit bus addresses 2^{32} locations; 2^{32} bytes $=4$ GiB.
7. 1.1^{n}: every exponential with base above 1 eventually beats every power (though here only for quite large n — the crossover is in the hundreds). $4\times10^{9}\approx2^{32}$, so about 32 steps.
8. $11=1011_{2}$. $n=11$, bit 1: result 5, square to 25. $n=5$, bit 1: result $5\times25=125$, square to 625. $n=2$, bit 0: square to 390,625. $n=1$, bit 1: result $125\times390\,625=48\,828\,125$. So $5^{11}=48\,828\,125$, with 3 squarings and 3 multiplications into the result (6 in all, 5 if the initial 1×5 is not counted), versus 10 naively.

Part II: Algebra

Chapter 4. The Language of Algebra: Variables, Expressions, and Equations

Algebra is arithmetic with the numbers left blank. This chapter rebuilds the vocabulary of that idea (variable, term, coefficient, expression, equation, identity, formula) and the handful of moves you make with it: substituting, expanding, collecting, factoring, and rearranging. Nothing here is hard in isolation. What makes algebra feel slippery after a long break is that the words have faded and the moves have become habits without reasons. We restore both, so that every later chapter can speak this language without stopping to translate.

Why it matters: every formula you will ever use (a loan repayment, a drug dose per kilogram, a spreadsheet column, a physics equation, a line of code) is an algebraic expression. Being fluent with expressions is what lets you use a formula you have never seen before, turn it round to answer the question you actually have, and spot when someone else's formula is wrong.

Where this chapter sits

You need: signed-number arithmetic and the order of operations (Chapter 1), fractions (Chapter 2), and exponent notation such as x^2 (Chapter 3).

This chapter unlocks: solving linear equations and inequalities (Chapter 5), systems of equations (Chapter 6), polynomials and factoring (Chapter 7), quadratics (Chapter 8), and, quietly, every formula in the rest of the book.

If you are in a hurry: try the Mastery check at the end first. If it is easy, skim the Refresher boxes for vocabulary and move on.

4.1. Why letters? Generalising arithmetic

4.1.1. The problem

Here is an old party trick. *Think of a number. Double it. Add ten. Halve the result. Subtract the number you first thought of.* Whatever number you chose, you are now holding 5.

Try it with 7: $7 \to 14 \to 24 \to 12 \to 5$. With 100: $100 \to 200 \to 210 \to 105 \to 5$. With -3: $-3 \to -6 \to 4 \to 2 \to 5$.

Three examples are not an explanation. How do we know it works for *every* number, including ones nobody will ever try? Arithmetic can only check one number at a time. We need a way to compute with "whatever number you thought of" as a single object.

4.1.2. Intuition

Give the unknown number a name. Call it n. Then do exactly the same operations, writing down what you have at each stage instead of a specific value:

Instruction	What you are holding
Think of a number	n
Double it	$2n$
Add ten	$2n + 10$
Halve the result	$n + 5$
Subtract the original number	5

The letter has done something no list of examples could do: it has carried out the arithmetic for *all numbers at once*. The halving step is where the magic happens: half of $2n + 10$ is $n + 5$, because halving each part of a sum halves the whole. Subtracting n then removes all trace of the original choice.

That is the entire spirit of algebra. A letter is a placeholder that lets you write down a calculation without committing to the input, so you can reason about the calculation itself.

4.1.3. Why it works

Arithmetic facts like $3 \times (4 + 5) = 3 \times 4 + 3 \times 5$ are true for specific numbers. But you will notice that the *shape* of such facts never depends on the particular numbers. Algebra promotes those shapes to first-class statements: $a(b + c) = ab + ac$ for all numbers a, b, c. Once a shape is established, you can use it anywhere, on numbers or on other expressions, without checking it again. The handful of rules that make this work (commutativity, associativity, distributivity, and the behaviour of 0 and 1) were reviewed in Chapter 1. Algebra is just those rules applied to symbols.



Variable

A **variable** is a symbol, usually a letter, that stands for a number we either do not know yet or want to leave unspecified. In $2n + 10$, n is the variable. The word comes from Latin *variabilis*, "changeable".



Constant

A **constant** is a quantity with a fixed value throughout a discussion. Literal numbers such as 10 or $-\frac{3}{4}$ are constants, and so are named numbers such as $\pi \approx 3.14159$ and $e \approx 2.71828$.



Why "algebra"?

The word comes from the title of a book written in Baghdad around 820 CE by Muhammad ibn Musa al-Khwarizmi: *al-Kitab al-mukhtasar fi hisab al-jabr wa'l-muqabala*, "The Compendious Book on Calculation by Restoring and Balancing". *Al-jabr* ("restoring") meant moving a subtracted quantity to the other side of an equation, where it becomes added. His name, Latinised as *Algoritmi*, gave us the word **algorithm**. Remarkably, al-Khwarizmi wrote everything in words: no letters, no symbols. Using letters for unknowns took another 750 years. François Viète (1590s) used vowels for unknowns and consonants for knowns, and René Descartes (1637) settled the convention we still use: x, y, z for unknowns, a, b, c for knowns.

US curriculum

Using letters to stand for numbers begins in **grades 6–7** ("expressions and equations" in the Common Core) and is the central theme of **Pre-Algebra** and **Algebra I** (grades 8–9).

4.2. The cast of characters: variables, constants, parameters, unknowns

4.2.1. The problem

Look at the equation of a straight line, which you will meet properly in Chapter 5:

$$y = mx + b$$

Four letters. Are they all "variables"? If you ask a mathematician, she will say x and y are variables, while m and b are… something else. Yet all four are letters standing for numbers. What is the difference, and why does it matter?

4.2.2. Intuition

The difference is not in the letters but in the *role* each letter plays in a particular discussion. A useful mental model: think of a family of programs.

- m and b are like configuration settings. Once chosen (say $m = 2, b = 1$), they stay fixed while you work with that particular line.
- x is the input that you vary freely.
- y is the output that depends on x.

Change m or b and you get a *different line*. Change x and you move *along the same line*.

Parameter

A **parameter** is a letter that is held fixed within one problem but could be changed to give a related problem. In $y = mx + b$, m and b are parameters: each choice of them gives one line from a whole family of lines. In $ax^2 + bx + c = 0$, the coefficients a, b, c are parameters.

Unknown

An **unknown** is a variable whose value we are trying to find from an equation. In $3x + 4 = 19$, x is the unknown, and the equation pins it down to $x = 5$. "Unknown" describes a variable's job in a problem rather than a different kind of object.

Independent and dependent variables

When one quantity is computed from another, the input is the **independent variable** and the output is the **dependent variable**. In $y = mx + b$, x is independent, y is dependent. In a formula such as $A = \pi r^2$, r is independent and A is dependent.

Roles can change. In the quadratic formula you will treat a, b, c as fixed and solve for x. But if you later ask "for which values of c does this equation have a solution?", c has become the variable of interest. Mathematicians switch roles constantly and rarely announce it. The context, and phrases like "for fixed a" or "solve for t", tell you who is playing which part.

Same letter, different jobs

Conventions are only conventions. Physics writes a for acceleration (a variable), statistics writes a for an intercept (a parameter), and e might be Euler's number or an error term. When you meet an unfamiliar formula, ask of each letter: *Is this fixed for the discussion, or does it vary? Am I solving for it?* Never assume from the letter alone.

Computer-science connection: parameters and arguments

The mathematical distinction maps neatly onto code. A function such as

```python
def line(x, m=2, b=1):
    return m * x + b
```

has an input x and two parameters with defaults. Calling `line(3)` moves along one fixed line; calling `line(3, m=5)` chooses a different member of the family. In machine learning the same split appears as *inputs* (the data you feed in) versus *parameters* or *weights* (the numbers training adjusts). Linear regression, which you will preview in Chapter 6 and meet properly in Volume 3, Chapter 31 , is literally the problem of choosing the parameters m and b that best fit observed (x, y) pairs. The roles flip: during training the data are fixed and m, b are the unknowns.

4.3. Anatomy of an expression: terms, factors, coefficients, like terms

4.3.1. The problem

To manipulate something you need names for its parts. A programmer who cannot say "argument", "return value", or "loop body" cannot discuss code. Algebra has the same need, and the vocabulary is small. Here is a typical expression, labelled:

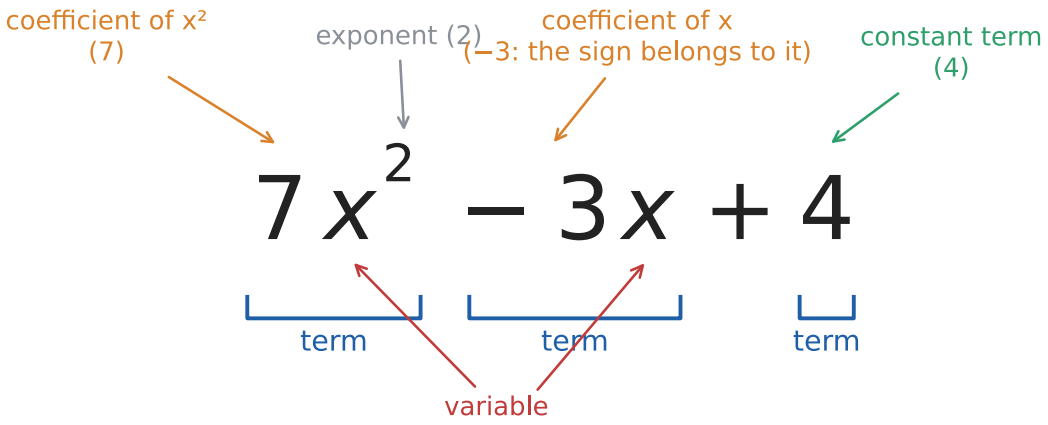


Figure 26. The parts of an expression

Expression

An **expression** is a meaningful combination of numbers, variables, and operations, with no equals sign: $7x^2 - 3x + 4$, $\frac{a+b}{2}$, $2\pi r$. An expression *has a value* once its variables are given values; it does not *assert* anything.

Term

The **terms** of an expression are the pieces that are added together. In $7x^2 - 3x + 4$ the terms are $7x^2$, $-3x$, and 4. Subtraction is read as adding a negative, so the minus sign travels with the term that follows it. Inside brackets things are grouped: $3(x + 2) - 4y$ has two terms at the top level, $3(x + 2)$ and $-4y$.

Factor

The **factors** of a term (or of a number) are the pieces that are multiplied together to make it. The term $6x^2y$ has factors 6, x, x, y (and combinations of them such as $2x$ or $3xy$). Terms are *added*; factors are *multiplied*. Keeping these two words apart avoids many errors.

Coefficient

A **coefficient** is the numerical factor multiplying the variable part of a term. In $7x^2 - 3x + 4$, 7 is the coefficient of x^2, and -3 is the coefficient of x (the sign is part of it). In x or $-y$ the coefficients are 1 and -1, usually not written. In ax^2, with a a parameter, we also call a the coefficient.

Constant term

The **constant term** is a term with no variable in it: the 4 in $7x^2 - 3x + 4$. If there is no such term, the constant term is 0.

Like terms

Like terms have exactly the same variable part, with each variable raised to the same power. $5x^2y$ and $-2yx^2$ are like terms (order of multiplication does not matter). $3x$ and $3x^2$ are *not* like terms; neither are xy and x. Only like terms can be combined into a single term.

4.3.2. Notation conventions worth restating

Algebra has a few writing conventions that are so ingrained nobody mentions them. After a gap they can trip you up.

- **Juxtaposition means multiplication.** $3x$ means $3 \times x$; ab means $a \times b$; $2(x + 1)$ means $2 \times (x + 1)$. This only works for letters: 23 is twenty-three, not 2×3, which is why you will see $2 \cdot 3$ with a raised dot when two numbers are multiplied.
- **Numbers come first, then letters, usually alphabetically.** Write $3xy$, not $x3y$ or $y3x$. They are equal, but the standard order makes like terms easy to spot.
- **Exponents bind tighter than everything else.** $3x^2$ means $3 \times (x^2)$, not $(3x)^2$. And $-x^2$ means $-(x^2)$, which is negative whenever $x \neq 0$. (See Chapter 3.)

- **A fraction bar groups.** $\frac{a+b}{2}$ means $(a+b)\div 2$, even though no brackets are written. Typing it into code you *must* add them: `(a + b) / 2`.
- **Order of operations** (brackets, exponents, multiplication and division, addition and subtraction) is exactly as in Chapter 1.



Degree of a term and of an expression

The **degree** of a term is the total number of variable factors in it: $7x^2$ has degree 2, $-3x$ has degree 1, 4 has degree 0, and $5x^2y$ has degree 3. The degree of a **polynomial** (a sum of such terms with whole-number powers) is the largest degree among its terms. Degree 1 means **linear**, degree 2 **quadratic**, degree 3 **cubic**. You will use these words constantly from Chapter 5 on.



The minus sign belongs to the term after it

In $5-2x$, the terms are 5 and $-2x$, so the coefficient of x is -2, not 2. This matters the moment you rearrange: $5-2x$ equals $-2x+5$, and it does *not* equal $2x-5$. If you reorder terms, each one carries its own sign with it.

US curriculum

The vocabulary of terms, factors, and coefficients is a **grade 6** Common Core standard (6.EE.A.2), and like terms are combined in **grade 7**. The word "degree" and the classification linear / quadratic / cubic arrive in **Algebra I**.

4.4. Expressions, equations, identities, and formulas

4.4.1. The problem

Compare these four lines:

$$\begin{aligned} &\text{(a)} \quad 2(x+3) \\ &\text{(b)} \quad 2(x+3)=14 \\ &\text{(c)} \quad 2(x+3)=2x+6 \\ &\text{(d)} \quad A=\ell w \end{aligned}$$

They look similar, but they are four different *kinds* of statement. Line (a) asserts nothing. Line (b) is true for exactly one value of x. Line (c) is true for every value of x. Line (d) describes a relationship between three quantities. Confusing these kinds is the root of a surprising number of algebraic blunders, so it pays to have precise names.



Equation

An **equation** is a statement that two expressions are equal: *left-hand side* (LHS) = *right-hand side* (RHS). An equation may be true or false depending on the values of its variables. Example: $2(x+3)=14$ is true when $x=4$ and false otherwise.

Solution, root, solution set

A **solution** of an equation is a value of the unknown that makes it true. For an equation of the form "expression $= 0$", a solution is also called a **root**. The **solution set** is the collection of all solutions, written with braces: the solution set of $x^2 = 9$ is $\{-3, 3\}$. An equation may have one solution, several, infinitely many, or none (solution set $\varnothing$, the empty set).

Identity

An **identity** is an equation that is true for *every* value of its variables, such as $2(x + 3) = 2x + 6$ or $(a + b)^2 = a^2 + 2ab + b^2$. Some books emphasise this with a triple bar, $2(x + 3) \equiv 2x + 6$, read "is identically equal to". Identities are the tools of algebra: each one licenses you to replace one expression by another.

Formula

A **formula** is an equation that expresses a rule or relationship between quantities, usually with one quantity isolated on the left, its **subject**: $A = \ell w$, $F = \frac{9}{5}C + 32$, $v = u + at$. A formula is a recipe: give values to the quantities on the right and it tells you the subject.

The same idea in table form:

Kind	Example	Has an equals sign?	True when?
Expression	$2(x + 3)$	No	Not a statement; it has a *value*
Conditional equation	$2(x + 3) = 14$	Yes	For some values (here only $x = 4$)
Identity	$2(x + 3) = 2x + 6$	Yes	For all values
Contradiction	$2(x + 3) = 2x + 7$	Yes	For no values
Formula	$A = \ell w$	Yes	Defines one quantity in terms of others

4.4.2. How to tell which kind you have

The reliable test is to simplify both sides and compare.

- $2(x + 3) = 2x + 6$: the left expands to $2x + 6$, identical to the right. Identity.
- $2(x + 3) = 2x + 7$: the left expands to $2x + 6$. Subtracting $2x$ from both sides leaves $6 = 7$, which is never true. Contradiction: no solutions.

- $2(x+3)=14$: the left is $2x+6$. Subtracting 6 gives $2x=8$, so $x=4$. Conditional: exactly one solution.

You will see these three outcomes again, under the names "one solution", "no solution", and "infinitely many solutions", for linear equations in Chapter 5 and for systems of equations in Chapter 6.



One example can refute an identity, but never prove one

Is $(x+1)^2=x^2+1$ an identity? Try $x=0$: both sides are 1. Looks good. Try $x=1$: the left is 4, the right is 2. It is *not* an identity.

One counterexample is enough to destroy a claimed identity. But no number of agreeing examples is enough to prove one, because there are infinitely many values you have not tried. To prove an identity, transform one side into the other using rules you already trust, as we do in Section 4.7.

Computer-science connection: testing identities at random points

Although examples cannot *prove* an identity, random testing is astonishingly good evidence for polynomial identities, and computer scientists exploit this. If two different polynomials of degree d are compared at a randomly chosen point, they can agree only if that point is one of at most d roots of their difference. Choose the point from a large set and the chance of being fooled is tiny. This is the **Schwartz–Zippel lemma**, and it underlies fast randomised algorithms for checking whether two programs or circuits compute the same thing.

PYTHON

```
import random

def probably_equivalent(f, g, trials=20):
    """Test whether two one-variable expressions agree at random points."""
    for _ in range(trials):
        x = random.uniform(-100, 100)
        if abs(f(x) - g(x)) > 1e-9 * max(1.0, abs(f(x))):
            return False, x          # a counterexample proves they differ
    return True, None                # strong evidence, not proof

print(probably_equivalent(lambda x: (x + 3)*(x + 2), lambda x: x**2 + 5*x + 6)[0])
print(probably_equivalent(lambda x: (x + 3)**2,      lambda x: x**2 + 9)[0])
```

```
True
False
```

Notice the tolerance in the comparison. Floating-point arithmetic rounds, so two mathematically equal expressions can differ in the last few bits. That is a theme of Volume 2, Chapter 32 .

US curriculum

Distinguishing expressions from equations is **grade 6**. Recognising equations with one, none, or infinitely many solutions is an explicit **grade 8** standard (8.EE.C.7a). The word "identity" is used more in **Algebra II** and **Precalculus** (trig identities).

4.5. Evaluating and substituting

4.5.1. The problem

A formula is only useful if you can feed values into it reliably. The rules are simple, yet substitution is where most sign errors in all of mathematics are born. Consider evaluating $3x^2 - 2xy + y^2$ when $x = -2$ and $y = 3$. A hurried calculation gives $-12 - 12 + 9 = -15$. The correct answer is 33. Where did the hurried version go wrong?

4.5.2. Intuition

Substitution means "replace each occurrence of the letter by the value". The danger is that a value is a *whole object*, while the expression around the letter may bind more tightly than you expect. x^2 with $x = -2$ means $(-2)^2 = 4$, not $-2^2 = -4$. The cure is mechanical: **always substitute inside brackets**. Brackets cost nothing and they protect the value.



Evaluate, substitute

To **evaluate** an expression is to compute its numerical value for given values of its variables. To **substitute** is to replace a variable by something else, a number or another expression. Evaluation is substitution followed by arithmetic.

Example 23. Procedure — Evaluating an expression by substitution

Input: an expression and a value for each variable in it.

Steps:

1. Rewrite the expression, replacing every occurrence of each variable by its value *in brackets*.
2. Evaluate powers (exponents) first.
3. Then multiplications and divisions, left to right.
4. Then additions and subtractions, left to right.
5. Check the sign of the result against a rough estimate.

Output: the numerical value of the expression.

4.5.3. Worked example

Evaluate $3x^2 - 2xy + y^2$ at $x = -2, y = 3$.

$$
\begin{aligned}
3x^2 - 2xy + y^2 &= 3(-2)^2 - 2(-2)(3) + (3)^2 && \text{(substitute, in brackets)} \\
&= 3(4) - 2(-2)(3) + 9 && \text{(powers first: } (-2)^2 = 4\text{)} \\
&= 12 - (-12) + 9 && \text{(multiply: } 2 \cdot (-2) \cdot 3 = -12\text{)} \\
&= 12 + 12 + 9 && \text{(subtracting a negative adds)} \\
&= 33.
\end{aligned}
$$

The hurried version made two errors: it squared -2 to get -4 (no brackets), and it lost the double negative in $-2xy$. Writing every bracket prevents both.

4.5.4. Where you'd use this

- **Health.** Body-mass index is $\text{BMI} = m/h^2$ (mass in kg, height in m). For 70 kg and 1.75 m: $70/(1.75)^2 = 70/3.0625 \approx 22.9$. The brackets matter: $70/1.75^2$ in a calculator is fine, but `=70/1.75*1.75` in a spreadsheet gives 70.
- **Physics.** A stone dropped from rest falls $s = \frac{1}{2}gt^2$ metres in t seconds. With $g = 9.8$ and $t = 3$: $\frac{1}{2}(9.8)(3)^2 = 4.9 \times 9 = 44.1$ m. Timing the fall of a stone down a well estimates its depth.
- **Computing.** Substitution at scale is what computers do. A spreadsheet fills a column by substituting each row's values into one formula; a database evaluates `price * qty * (1 + tax_rate)` for millions of rows; a GPU shader evaluates the same lighting expression for every pixel of every frame. Each is the procedure above, run very fast.

4.5.5. Substituting an expression

You can substitute an expression, not just a number. This is how one formula is fed into another. If $y = x^2 + 2x$ and $x = t - 1$, then

$$
\begin{aligned}
y &= (t-1)^2 + 2(t-1) && \text{(replace } x \text{ by } (t-1)\text{)} \\
&= (t^2 - 2t + 1) + (2t - 2) && \text{(expand each bracket; see below)} \\
&= t^2 - 1. && \text{(collect like terms)}
\end{aligned}
$$

The brackets around $t - 1$ are not optional: without them, x^2 would become $t - 1^2$, which is a different expression entirely. Substitution of expressions is the algebraic face of *function composition*, which you will meet in Chapter 11.



The square of a negative

With $x = -3$:

- $x^2 = (-3)^2 = 9$;
- $-x^2 = -(x^2) = -(9) = -9$;
- $(-x)^2 = (3)^2 = 9$.

So $-x^2$ and $(-x)^2$ are different expressions, and $-x^2 + (-x)^2 = 0$ for every x. Spreadsheets get this "wrong": in Excel, `=-A1^2` with `A1 = 3` gives `9`, because Excel applies the unary minus before the power. Python and mathematics agree: `-3**2` is `-9`.

Try it: −x² in Excel and Python .

4.5.6. Expression trees: what the order of operations really is

The order of operations can feel like an arbitrary rule to memorise. It is better understood as a description of a *tree*. Every expression has a hidden structure in which each operation combines exactly two sub-expressions. Here is $3(x+2)-4y$:

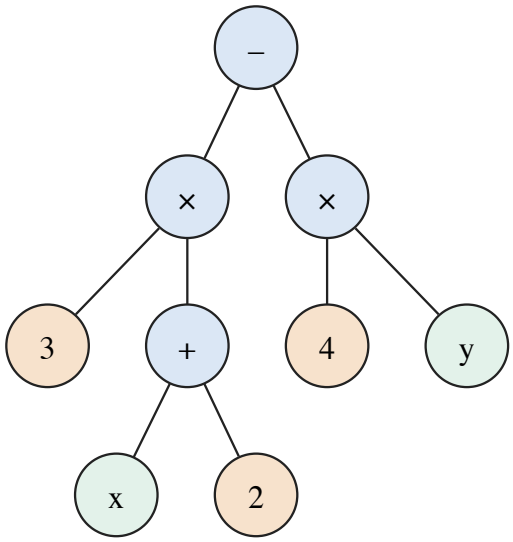


Figure 27. The expression tree of $3(x+2)-4y$

To evaluate, start at the leaves (numbers and variables), replace each variable by its value, and work upward: each internal node applies its operation to the values of its children. The root's value is the value of the expression. "Brackets first, then exponents, then multiplication, then addition" is simply the convention for how a flat line of symbols encodes such a tree. Brackets exist to override the default tree shape. *Try it:* Draw the tree of 3(x+2)−4y .

Substitution is now easy to picture: replace a leaf labelled x by the value, or by an entire subtree.

Computer-science connection: parsing and expression trees

Every compiler and interpreter turns the text `3*(x + 2) - 4*y` into exactly this tree (an **abstract syntax tree**, AST) before doing anything else. The order of operations is encoded as operator **precedence** in the parser's grammar. Python will show you its tree:

```python
import ast

src = "3*(x + 2) - 4*y"
tree = ast.parse(src, mode="eval")
print(ast.dump(tree.body, indent=1))
print(eval(compile(tree, "<expr>", "eval"), {"x": 5, "y": 2}))
```

```
BinOp(
 left=BinOp(
  left=Constant(value=3),
  op=Mult(),
  right=BinOp(
   left=Name(id='x', ctx=Load()),
   op=Add(),
   right=Constant(value=2))),
 op=Sub(),
 right=BinOp(
  left=Constant(value=4),
  op=Mult(),
  right=Name(id='y', ctx=Load())))
13
```

Evaluating a tree is a short recursive function, and the place where a variable's value is looked up is precisely substitution:

```python
def evaluate(node, env):
    """Evaluate an expression tree.

    A node is a number, a variable name (str), or a tuple (op, left, right).
    """
    if isinstance(node, (int, float)):
        return node
    if isinstance(node, str):
        return env[node]                # substitution happens here
    op, left, right = node
    a, b = evaluate(left, env), evaluate(right, env)
    if op == "+": return a + b
    if op == "-": return a - b
    if op == "*": return a * b
    if op == "/": return a / b
    raise ValueError(f"unknown operator {op!r}")

# 3(x + 2) - 4y  as a tree
tree = ("-", ("*", 3, ("+", "x", 2)), ("*", 4, "y"))
print(evaluate(tree, {"x": 5, "y": 2}))    # 3*7 - 8 = 13
```

```
13
```

A **computer algebra system** (CAS) such as Mathematica, Maple, or Python's SymPy library goes one step further: instead of evaluating the tree to a number, it *rewrites* the tree using identities such as the distributive law. Expanding, factoring, and simplifying are all tree transformations. That is exactly what the next three sections teach you to do by hand.



Build-a-tree

Type an expression and see its tree drawn live. Drag values onto the variable leaves and watch each internal node light up with its value as evaluation flows from leaves to root. Then remove a pair of brackets and see how the tree, and the answer, changes.

4.6. Exercises: vocabulary, kinds of statement, substitution

1. In $5a^2 - a + 7 - 2ab$, list the terms, and give the coefficient of a, the coefficient of ab, and the constant term.
2. Classify each as an expression, conditional equation, identity, contradiction, or formula: (a) $3(x-1) = 3x - 3$; (b) $3(x-1) = 3x + 3$; (c) $3(x-1) = 12$; (d) $A = \pi r^2$; (e) $4x - 7$.
3. Evaluate $2a^2 - 3ab + b^3$ when $a = -1$, $b = 2$.
4. Evaluate $-x^2 + (-x)^2$ when $x = 4$. What happens for other values of x, and why?
5. Substitute $x = t - 1$ into $x^2 + 2x + 1$ and simplify. Explain the result in one sentence.
6. Find one value of x that shows $(x+1)^2 = x^2 + 1$ is not an identity. Are there any values for which it *is* true?
7. Draw the expression tree for $\frac{a+b}{2} - c^2$. Which operation is at the root?

▼ **Answers**

1. Terms: $5a^2$, $-a$, 7, $-2ab$. Coefficient of a is -1 (the sign belongs to the term, and the 1 is invisible). Coefficient of ab is -2. Constant term 7.
2. (a) Identity: the left expands to $3x - 3$. (b) Contradiction: the left is $3x - 3$, and $3x - 3 = 3x + 3$ reduces to $-3 = 3$. (c) Conditional equation: $3x - 3 = 12$ gives $3x = 15$, $x = 5$. (d) Formula (subject A). (e) Expression.
3. $2(-1)^2 - 3(-1)(2) + (2)^3 = 2(1) - (-6) + 8 = 2 + 6 + 8 = 16$.
4. $-(4^2) + (-4)^2 = -16 + 16 = 0$. It is 0 for every x, because $(-x)^2 = x^2$ (a negative times a negative is positive), so the expression is $-x^2 + x^2$.
5. $(t-1)^2 + 2(t-1) + 1 = (t^2 - 2t + 1) + (2t - 2) + 1 = t^2$. The original expression is $(x+1)^2$, and $x + 1 = t$, so of course it becomes t^2.

6. $x = 1$: the left is 4, the right is 2. Expanding, $(x+1)^2 = x^2 + 2x + 1$, so the equation is $x^2 + 2x + 1 = x^2 + 1$, i.e. $2x = 0$: it is true only for $x = 0$.

7. The root is the subtraction. Its left child is a division node (children: an addition node with leaves a, b; and the leaf 2). Its right child is a power node with leaves c and 2. The fraction bar acts as brackets around $a + b$.

4.7. Expanding brackets: the distributive law

4.7.1. The problem

Quick: what is 7×38? Most people who can do this in their heads do it as $7 \times 40 - 7 \times 2 = 280 - 14 = 266$. They have used, perhaps without naming it, the single most important law in algebra. Every time you "multiply out brackets", you are using it. Every time you factorise, you are using it backwards. Let us make it explicit and see *why* it is true.

4.7.2. Intuition: the area model

Think of multiplication as the area of a rectangle. A rectangle of height 3 and width $x + 2$ can be cut into two rectangles: one 3 by x, one 3 by 2. The total area is unchanged by cutting, so $3(x + 2) = 3x + 6$.

The same picture handles two brackets. A rectangle of width $x + 2$ and height $x + 3$ cuts into four pieces, one for each pairing of a width-part with a height-part:

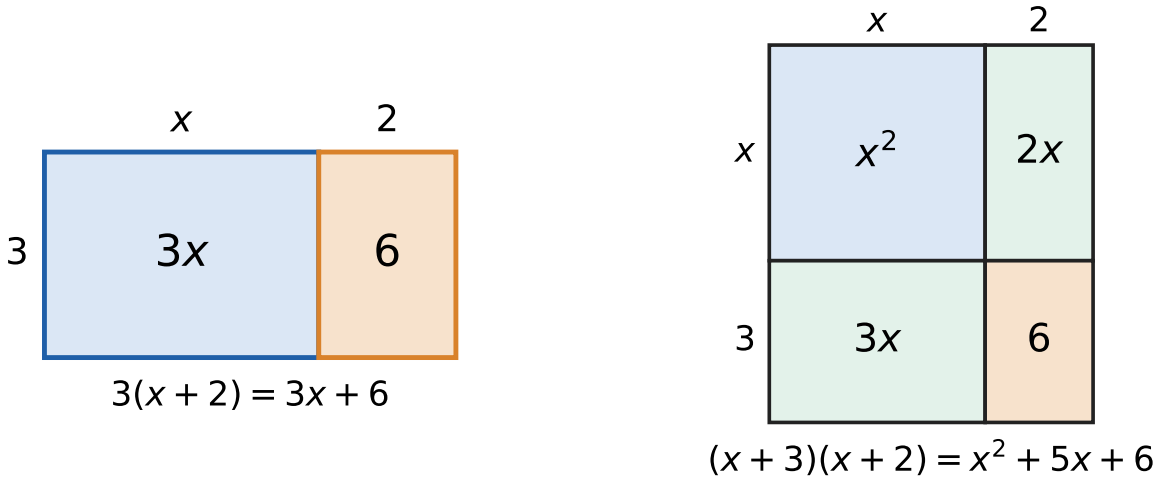


Figure 28. The area model of the distributive law

Adding the four areas gives $x^2 + 2x + 3x + 6 = x^2 + 5x + 6$. The picture explains the rule you may remember as "multiply everything in the first bracket by everything in the second".

4.7.3. Why it works

The **distributive law** is one of the basic properties of numbers:

$$a(b + c) = ab + ac \qquad \text{and} \qquad (b + c)a = ba + ca.$$

It holds for every number, positive or negative, whole or fractional; the area picture is a convincing illustration for positive lengths, and the rules of signed arithmetic in Chapter 1 extend it to all numbers. Because $b - c = b + (-c)$, it also covers subtraction: $a(b - c) = ab - ac$. And because it holds for *any* a, we may let a itself be a bracketed expression. That gives two-bracket expansion with no new rule at all:

$$\begin{aligned}(x+3)(x+2) &= (x+3)\,x + (x+3)\cdot 2 && \text{(distribute the whole } (x+3) \text{ over } x+2)\\ &= x^2 + 3x + 2x + 6 && \text{(distribute again, twice)}\\ &= x^2 + 5x + 6. && \text{(collect like terms)}\end{aligned}$$



Distributive law, expand

The **distributive law** says multiplication "distributes over" addition: $a(b + c) = ab + ac$. To **expand** (or "multiply out") an expression is to use this law to remove brackets, producing a sum of terms. The opposite move, rewriting a sum as a product, is **factoring** (Section 4.9).



Binomial, product

A **binomial** is an expression with two terms, such as $x + 3$ or $2a - 5b$ (*bi* = two, *nomen* = name). A **monomial** has one term, a **trinomial** three. A **product** is the result of a multiplication; "the product of $x + 3$ and $x + 2$" means $(x + 3)(x + 2)$.

Example 24. Procedure — Expanding a product of brackets

Input: a product of two (or more) bracketed sums.

Steps:

1. Multiply *each term* of the first bracket by *each term* of the second, carrying signs with the terms. If the brackets have m and n terms, you will get $m \times n$ products. Count them.
2. Simplify each product using the exponent law $x^a x^b = x^{a+b}$ and sign rules.
3. Collect like terms (Section 4.8).
4. For three or more brackets: expand two of them first, then multiply the result by the next bracket.

Output: an equivalent expression with no brackets, a sum of terms.

4.7.4. Worked examples

Example 1. A negative multiplier. Expand $-3(2x - 5)$.

$$\begin{aligned}-3(2x-5) &= (-3)(2x) + (-3)(-5) && \text{(each term, with its sign, times } -3)\\ &= -6x + 15.\end{aligned}$$

Example 2. Two binomials with signs. Expand $(2x - 1)(x + 4)$. There are $2 \times 2 = 4$ products:

$$\begin{aligned}(2x-1)(x+4) &= (2x)(x)+(2x)(4)+(-1)(x)+(-1)(4) && \text{(each term by each term)}\\ &= 2x^2+8x-x-4 && \text{(simplify each product)}\\ &= 2x^2+7x-4. && \text{(collect like terms: } 8x-x=7x\text{)}\end{aligned}$$

Check by substitution. With $x=1$: the original is $(1)(5)=5$; the expansion is $2+7-4=5$. Agreement at one point does not prove the expansion right (see the Watch out in Section 4.4), but disagreement would prove it wrong, so this is a quick, worthwhile sanity check.

Example 3. A binomial times a trinomial. Expand $(x+1)(x^2-x+1)$. There are $2\times 3=6$ products:

$$\begin{aligned}(x+1)(x^2-x+1) &= x\cdot x^2+x(-x)+x\cdot 1+1\cdot x^2+1(-x)+1\cdot 1\\ &= x^3-x^2+x+x^2-x+1\\ &= x^3+1. && \text{(the middle terms cancel in pairs)}\end{aligned}$$

That cancellation is not luck: it is the "sum of cubes" factorisation $x^3+1=(x+1)(x^2-x+1)$, which you will meet again in Chapter 7.

4.7.5. Special products worth knowing by sight

Three expansions appear so often that it is worth recognising them instantly, in both directions.

$$\begin{aligned}(a+b)^2 &= a^2+2ab+b^2 && \text{(square of a sum)}\\ (a-b)^2 &= a^2-2ab+b^2 && \text{(square of a difference)}\\ (a+b)(a-b) &= a^2-b^2 && \text{(difference of two squares)}\end{aligned}$$

Each one is just the four-product rule. For the first:
$(a+b)(a+b)=a^2+ab+ba+b^2=a^2+2ab+b^2$. For the third, the cross terms $-ab$ and $+ab$ cancel. In the area model, $(a+b)^2$ is a square of side $a+b$ cut into an a^2 square, a b^2 square, and *two* ab rectangles.



The freshman's dream

$(a+b)^2\neq a^2+b^2$. Squaring does *not* distribute over addition. With $a=3, b=4$: $(3+4)^2=49$ but $3^2+4^2=25$. The missing $2ab=24$ is the two rectangles in the area picture. The same error in other clothing: $\sqrt{a+b}\neq\sqrt{a}+\sqrt{b}$ and $\frac{1}{a+b}\neq\frac{1}{a}+\frac{1}{b}$. Only multiplication distributes over addition.



Mental arithmetic with special products

The difference of two squares turns certain multiplications into subtractions: $49\times 51=(50-1)(50+1)=2500-1=2499$, and $97\times 103=10000-9=9991$. The square of a sum does squares: $103^2=10000+600+9=10609$. Before electronic calculators, tables of squares were used to multiply via the identity $ab=\frac{(a+b)^2-(a-b)^2}{4}$ ("quarter-squares multiplication"), and the same trick appears in some hardware multipliers.

4.7.6. Alternative approaches and choosing a method

- **FOIL** (First, Outer, Inner, Last) is a mnemonic for multiplying two binomials: $(a+b)(c+d) = ac + ad + bc + bd$. It is fine for exactly that case, but it is just "each term by each term" with an order attached, and it gives no help with a trinomial. Many teachers now discourage it for that reason.
- **The grid (box) method** is the area model written as a table: write one bracket's terms across the top, the other's down the side, fill each cell with a product, then add all the cells. It scales to any number of terms and makes it hard to miss a product. It is the best choice when brackets have three or more terms.
- **Recognising a special product** is fastest when it applies. Look for squares and for pairs like $(a+b)(a-b)$ before reaching for the general method.

Rule of thumb: spot a special product if you can; use "each by each" (or FOIL) for two binomials; use a grid for anything larger.



Resize the rectangle

Drag sliders for a and b and watch a square of side $a+b$ split into a^2, b^2, and two ab rectangles. Toggle "freshman's dream" to overlay $a^2 + b^2$ alone and see exactly what area is missing.

4.7.7. Where you'd use this

- **Interest on interest.** £1000 at 5% for two years becomes $1000(1+r)^2 = 1000(1 + 2r + r^2)$ with $r = 0.05$: $1000(1 + 0.10 + 0.0025) = £1102.50$. The expansion shows where the money comes from: $2r$ is two years of simple interest (£100), and r^2 is the interest earned on the first year's interest (£2.50).
- **Home and garden.** A square lawn of side x metres gets a path 2 m wide all round. The outer square has side $x + 4$, so the path's area is $(x+4)^2 - x^2 = x^2 + 8x + 16 - x^2 = 8x + 16$ square metres. For a 10 m lawn that is 96 m², which tells you how many paving slabs to buy.
- **Mental arithmetic.** $23 \times 17 = (20+3)(20-3) = 400 - 9 = 391$, the difference of two squares in action.

4.8. Collecting like terms and simplifying

4.8.1. The problem

Expanding usually leaves a mess: $4(2x - y) - 3(x - 2y) + 5$ becomes a string of six terms. We want the simplest equivalent form. What does "simplest" mean, and which combinations are legal?

4.8.2. Intuition and why it works

$3x + 5x = 8x$ because three apples plus five apples is eight apples. But the real justification is the distributive law read backwards:

$$3x + 5x = (3+5)x = 8x.$$

That explains why only *like* terms combine. $3x + 5x^2$ has no common variable part to pull out, so there is nothing to add: it is already as simple as it gets. Apples and apple-squares do not merge.



Simplify

To **simplify** an expression is to rewrite it in an equivalent form that is shorter or more useful, typically with brackets removed, like terms collected, and terms written in a standard order (usually decreasing powers: $2x^2 + 7x - 4$). "Simplest" depends on purpose: sometimes the factored form is the more useful one.

Example 25. Procedure — Collecting like terms

Input: an expression.

Steps:

1. Expand any brackets. When a bracket is preceded by a minus sign, negate *every* term inside it.
2. Group like terms together (underline them, or reorder them, each term keeping its sign).
3. Add the coefficients within each group; keep the variable part unchanged.
4. Write the result in standard order and drop any terms with coefficient 0.

Output: an equivalent expression with no two like terms.

4.8.3. Worked example

Simplify $4(2x - y) - 3(x - 2y) + 5$.

$$\begin{aligned} 4(2x - y) - 3(x - 2y) + 5 &= 8x - 4y - 3x + 6y + 5 && \text{(expand; } (-3)(-2y) = +6y\text{)} \\ &= (8x - 3x) + (-4y + 6y) + 5 && \text{(group like terms)} \\ &= 5x + 2y + 5. && \text{(add coefficients)} \end{aligned}$$

Check at $x = 1, y = 1$: original $4(1) - 3(-1) + 5 = 4 + 3 + 5 = 12$; result $5 + 2 + 5 = 12$. Consistent.



Subtracting a bracket negates everything inside

$10 - (x - 3)$ is $10 - x + 3 = 13 - x$, not $10 - x - 3$. A minus sign in front of a bracket is a hidden coefficient of -1, and it must be distributed to *every* term inside. This single error accounts for a large fraction of all wrong answers in school algebra, and in hand-derived formulas in engineering too.



Is it simpler or just different?

"Simplify" is not a precise instruction. $x^2 - 1$ and $(x-1)(x+1)$ are equally simple. If you want to solve $x^2 - 1 = 0$, the factored form is better; if you want to add it to $3x^2 + 2$, the expanded form is better. Choose the form that serves your next step.

4.9. Factoring out a common factor

4.9.1. The problem

Expanding turns products into sums. Often we need to go the other way. Suppose a shop sells items at 12 dollars, 18 dollars, and 30 dollars, and you buy n of each: the total $12n + 18n + 30n$ is obviously $60n$. Less obviously, $6x^2 + 9x$ can be rewritten as $3x(2x + 3)$. Why would you want to? Because a product tells you things a sum hides. A product is zero exactly when one of its factors is zero, so $3x(2x + 3) = 0$ immediately gives $x = 0$ or $x = -\frac{3}{2}$; the sum $6x^2 + 9x = 0$ tells you nothing at a glance. Products also cancel in fractions, and sums do not.

4.9.2. Intuition

Factoring is the distributive law run backwards: $ab + ac = a(b + c)$. Look for something every term shares, pull it out to the front, and write what is left of each term inside a bracket. In the area picture, you are taking several rectangles of the same height and pushing them together into one long rectangle.



Factor, factorise, greatest common factor

To **factor** (US) or **factorise** (UK) an expression is to rewrite it as a product. The **greatest common factor** (GCF), also called the highest common factor (HCF), of several terms is the largest expression that divides all of them: the greatest common divisor of the coefficients times each shared variable raised to its *lowest* power. The GCF of $12x^3y$, $-18x^2y^2$, and $6x^2y$ is $6x^2y$.

Example 26. Procedure — Factoring out the greatest common factor

Input: a sum of terms.

Steps:

1. Find the GCD of the numerical coefficients (see Chapter 1 and Volume 3, Chapter 23).
2. For each variable that appears in *every* term, take the lowest power that occurs.
3. The GCF is the product of steps 1 and 2. (If the leading term is negative, you may choose to include a factor of -1.)
4. Divide each term by the GCF; the quotients form the bracket.
5. Write GCF $\times$ (bracket). **Check by expanding.** The bracket should have the same number of terms as the original, and its terms should have no common factor left.

Output: an equivalent product, GCF $\times$ (sum).

4.9.3. Worked example

Factor $12x^3y - 18x^2y^2 + 6x^2y$.

- Coefficients $12, -18, 6$: GCD is 6.
- x appears as x^3, x^2, x^2: lowest power x^2.
- y appears as y, y^2, y: lowest power y.
- GCF $= 6x^2y$.

$$\begin{aligned} 12x^3y - 18x^2y^2 + 6x^2y &= 6x^2y\left(\frac{12x^3y}{6x^2y} - \frac{18x^2y^2}{6x^2y} + \frac{6x^2y}{6x^2y}\right) \qquad \text{(divide each term by the GCF)} \\ &= 6x^2y\,(2x - 3y + 1). \end{aligned}$$

Check: $6x^2y \cdot 2x = 12x^3y$, $6x^2y \cdot (-3y) = -18x^2y^2$, $6x^2y \cdot 1 = 6x^2y$. Correct, and three terms in, three terms out.



The vanishing 1

When the whole of a term is the GCF, it leaves 1 behind, not nothing. $6x^2y$ divided by $6x^2y$ is 1, so the bracket is $(2x - 3y + 1)$. Writing $6x^2y(2x - 3y)$ loses a term, and expanding to check catches the mistake instantly.

Factoring out a negative. $-4x + 8$ can be written $4(-x + 2)$ or $-4(x - 2)$. Both are correct. The second is usually preferred because it makes the bracket's leading coefficient positive, which makes common brackets easier to spot, for instance in $x(x - 2) - 4(x - 2) = (x - 4)(x - 2)$. That last move, pulling out a common *bracket*, is the seed of factoring by grouping in Chapter 7.

4.9.4. Where you'd use this

- **Shopping.** Prices of £12, £18, and £30 plus 20% VAT on each: $1.2 \times 12 + 1.2 \times 18 + 1.2 \times 30 = 1.2(12 + 18 + 30) = 1.2 \times 60 = £72$. Factoring is why a till can add up the items first and apply tax once.
- **Physics.** A ball thrown upward at 20 m/s is at height $h = 20t - 4.9t^2$ metres after t seconds. Factoring gives $h = t(20 - 4.9t)$: a product is zero exactly when one of its factors is, so the ball is at ground level at $t = 0$ and at $t = 20/4.9 \approx 4.1$ s. That "factor, then set each factor to zero" move is the main method of Chapter 8.
- **Computing.** Factored forms do fewer multiplications (Horner's method, below), and factoring a common term out of a loop (`total = rate * sum(xs)` instead of `sum(rate * x for x in xs)`) is a standard optimisation.

4.9.5. Where factoring goes next

Taking out a common factor is the first and simplest kind of factoring. Recognising $x^2 - 9$ as $(x - 3)(x + 3)$ (difference of two squares) and $x^2 + 5x + 6$ as $(x + 2)(x + 3)$ are harder skills developed in Chapter 7 and Chapter 8. The habit to form now: when faced with a sum, always ask first, "is there a common factor?"

Computer-science connection: factoring saves multiplications (Horner's method)

Factoring is not only for solving equations; it changes how much work evaluation costs. Take $2x^3 - 3x^2 + 5x - 7$. Evaluated term by term it needs several multiplications for the powers. Now factor x out of the first three terms, and again, and again:

$$2x^3 - 3x^2 + 5x - 7 = \big((2x - 3)x + 5\big)x - 7.$$

This **nested** form (**Horner's method**) needs only one multiplication and one addition per coefficient, and it is how numerical libraries evaluate polynomials.

PYTHON

```
def horner(coeffs, x):
    """Evaluate a polynomial given coefficients from highest power down."""
    result = 0
    for c in coeffs:
        result = result * x + c     # factor out x, again and again
    return result

# 2x^3 - 3x^2 + 5x - 7 at x = 2
print(horner([2, -3, 5, -7], 2))
print(2*2**3 - 3*2**2 + 5*2 - 7)
```

```
7
7
```

Compilers perform similar algebra on your code: rewriting `a*b + a*c` as `a*(b + c)` saves a multiplication. (For floating-point numbers the two forms can round differently, which is why compilers only do this when told they may; see Volume 2, Chapter 32 .)

US curriculum

The distributive property and factoring out a common factor (such as $12 + 18 = 6(2 + 3)$) appear in **grade 6**; expanding and collecting with variables in **grade 7**. Multiplying binomials, special products, and the GCF of monomials are core **Algebra I** topics (grades 8–9), leading directly into factoring quadratics.

4.10. Exercises: expanding, collecting, factoring

1. Expand $-4(3 - 2x)$.
2. Expand and simplify $(x - 5)(x + 3)$.
3. Expand $(3a - 2b)^2$ using a special product.
4. Simplify $5(2x - 3) - 2(4x - 7)$.

5. Factor $15x^2 - 10x$.
6. Factor $8a^2b - 12ab^2 + 4ab$ completely.
7. Use special products to compute 103^2 and 97×103 without a calculator.
8. Prove that the sum of any three consecutive integers is divisible by 3. (Let the smallest be n.)
9. Expand $(x - 2)(x^2 + 2x + 4)$. What pattern from the chapter does the answer match?

▼ **Answers**

1. $(-4)(3) + (-4)(-2x) = -12 + 8x$, usually written $8x - 12$.
2. $x \cdot x + 3x - 5x - 15 = x^2 - 2x - 15$.
3. $(a - b)^2 = a^2 - 2ab + b^2$ with $a \to 3a$, $b \to 2b$:
   $(3a)^2 - 2(3a)(2b) + (2b)^2 = 9a^2 - 12ab + 4b^2$.
4. $10x - 15 - 8x + 14 = 2x - 1$. Note $(-2)(-7) = +14$.
5. GCF $5x$: $5x(3x - 2)$. Check: $15x^2 - 10x$.
6. GCF of coefficients 4; lowest powers a^1, b^1. GCF $4ab$: $4ab(2a - 3b + 1)$. Don't lose the 1.
7. $103^2 = (100 + 3)^2 = 10000 + 600 + 9 = 10609$.
   $97 \times 103 = (100 - 3)(100 + 3) = 10000 - 9 = 9991$.
8. $n + (n + 1) + (n + 2) = 3n + 3 = 3(n + 1)$. Since $n + 1$ is an integer, the sum is 3 times an integer. (This is a genuine proof: one algebraic computation covers every case, as in the "think of a number" trick.)
9. $x^3 + 2x^2 + 4x - 2x^2 - 4x - 8 = x^3 - 8$. It is the difference-of-cubes cousin of Example 3:
   $a^3 - b^3 = (a - b)(a^2 + ab + b^2)$ with $a = x$, $b = 2$.

4.11. Equations and the balance model

4.11.1. The problem

So far we have transformed *expressions* into equivalent expressions. With *equations* we want something different: to find the values that make the equation true. The strategy is to transform the equation step by step into something so simple that the answer can be read off, such as $x = 4$. But which transformations are allowed? If you change an equation, how do you know the new one has the same solutions?

4.11.2. Intuition: the balance

Picture an old-fashioned pan balance. An equation says the two pans weigh the same. Whatever you do to one pan, you must do to the other, and the balance stays level. Below, each blue box is an unknown weight x and each orange disc weighs 1 unit.

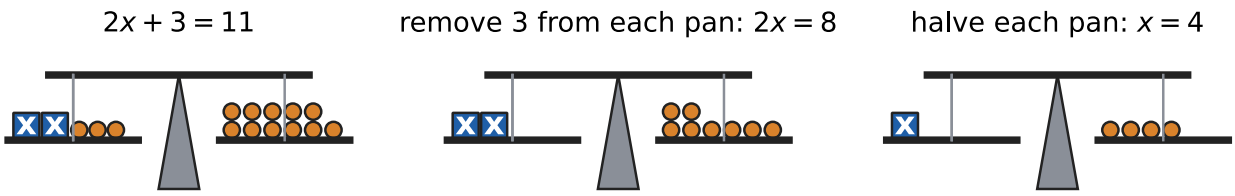


Figure 29. Solving $2x + 3 = 11$ on a balance

Remove three discs from each side: the pans still balance, so $2x = 8$. Halve the contents of each pan: still level, so $x = 4$. **Check:** $2(4) + 3 = 11$. Correct.

This is al-Khwarizmi's "balancing", and it is the whole idea of equation solving. Chapter 5 develops it into a complete algorithm.

4.11.3. Why it works: equivalent equations



Equivalent equations

Two equations are **equivalent** if they have exactly the same solution set. $2x + 3 = 11$, $2x = 8$, and $x = 4$ are equivalent: each is true for $x = 4$ and for nothing else. Solving an equation means replacing it by a chain of equivalent equations, ending with one whose solution is obvious.

Which operations produce an *equivalent* equation? The key requirement is that the step can be **undone**. If $A = B$ then certainly $A + 5 = B + 5$; and conversely, if $A + 5 = B + 5$ we can subtract 5 to recover $A = B$. Because we can go both ways, no solutions are gained or lost. The safe operations are:

- add or subtract the same quantity on both sides (always reversible);
- multiply or divide both sides by the same **non-zero** quantity (reversible: undo by dividing or multiplying);
- replace either side by an equivalent *expression* (expanding, collecting, factoring);
- swap the two sides.



Operations that are not reversible

Multiplying by zero destroys information: $x = 3$ times 0 gives $0 = 0$, which every x satisfies. **Dividing by an expression that might be zero** can lose solutions: dividing $x^2 = 5x$ by x gives $x = 5$, silently losing the solution $x = 0$. Factor instead: $x^2 - 5x = x(x - 5) = 0$, so $x = 0$ or $x = 5$.

Squaring both sides can create solutions: $x = 3$ squared gives $x^2 = 9$, which also has $x = -3$. Squaring is sometimes necessary (equations with square roots), but then every candidate solution must be checked in the original equation. Such impostors are called **extraneous solutions**.



Equals means "is the same as", not "the answer is"

Many people learned to treat $=$ as a button meaning "now write the answer", and so write chains such as $3 + 4 = 7 \times 2 = 14$. Each $=$ in a chain must be a true statement on its own, and $3 + 4 = 7 \times 2$ is false. In algebra, $=$ is a symmetric relation: $A = B$ and $B = A$ say the same thing, and both sides are on equal footing. Keep chains of equalities honest and your derivations will be checkable line by line.



Balance builder

Load an equation onto a virtual balance as boxes (unknowns) and discs (units), including "negative" balloons that pull upward. Apply operations to both pans and watch the beam. Try an illegal move, such as removing a disc from one pan only, and watch it tip. Try "multiply both pans by 0" and watch every possible x become a solution.

4.11.4. Where you'd use this

- **Small business.** A cake stall costs £40 to set up; each cake costs £0.50 to make and sells for £2. Breaking even after n cakes means $40 + 0.5n = 2n$. Subtract $0.5n$ from both pans: $40 = 1.5n$, so $n = 40/1.5 \approx 26.7$. You need to sell 27 cakes before you make any profit.
- **Everywhere else.** Every "how many/how long/how much" question with a known formula is an equation to solve: how many months to save for a £600 phone at £45 a month ($45m = 600$, $m \approx 13.3$, so 14 months); what dose gives a target blood concentration; how many servers handle a forecast load. Chapter 5 turns this into a systematic method.

4.12. Rearranging formulas: changing the subject

4.12.1. The problem

The formula $F = \frac{9}{5}C + 32$ converts a Celsius temperature to Fahrenheit. A thermostat in a US house reads $68°\text{F}$; what is that in Celsius? You could substitute $F = 68$ and solve the resulting equation. But if you need to convert a thousand readings, or write a function `f_to_c`, you want a *formula* with C as its subject. Rearranging is equation solving where the "answer" is an expression rather than a number.

4.12.2. Intuition: do and undo

Read the formula as a recipe that builds F from C: start with C, *multiply by* $\frac{9}{5}$, then *add* 32. To get back from F to C, undo the steps *in reverse order*, each with its inverse operation: *subtract* 32, then *divide by* $\frac{9}{5}$ (that is, multiply by $\frac{5}{9}$).

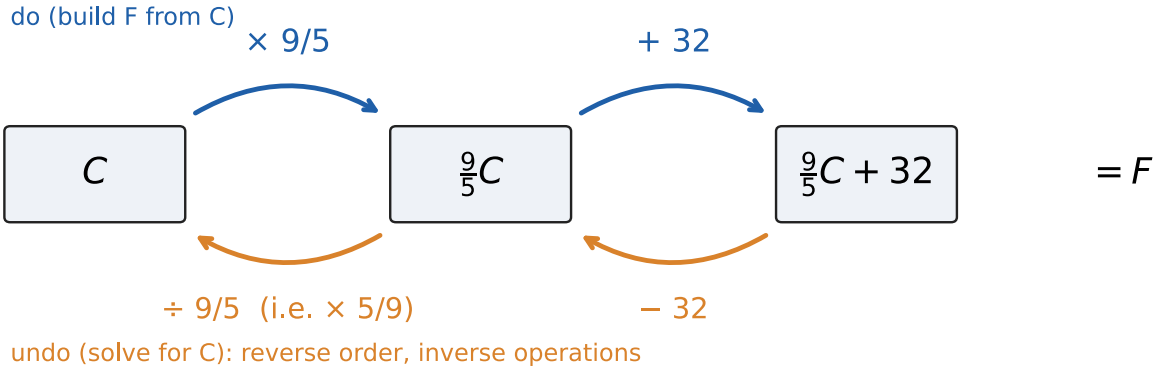


Figure 30. Changing the subject: undo the operations in reverse order

It is the same logic as getting undressed: you put on socks and then shoes, so you take off shoes and then socks. Inverses in reverse order.



Subject of a formula, inverse operation

The **subject** of a formula is the variable standing alone on one side, usually the left: F is the subject of $F = \frac{9}{5}C + 32$. **Changing the subject** (also **rearranging**, **transposing**, or **solving for** a variable) means producing an equivalent formula with a different subject. An **inverse operation** undoes another: addition and subtraction are inverses, as are multiplication and division (by a non-zero number), and squaring and taking the square root (with care about signs).

Example 27. Procedure — Changing the subject of a formula

Input: a formula and the variable v that should become the subject.

Steps:

1. Clear fractions by multiplying both sides by any denominators (noting which quantities must be non-zero).
2. Expand brackets that contain v.
3. Gather all terms containing v on one side and all other terms on the other side (add or subtract terms on both sides).
4. If v appears in more than one term, factor it out: $v(\dots)$.
5. Divide both sides by the coefficient of v (state that it must be non-zero).
6. If v appears inside a power or root, apply the inverse (root or power) last, considering $\pm$.
7. **Check:** substitute a convenient set of numbers into both the original and the new formula.

Output: a formula $v = \dots$ with v absent from the right-hand side, plus any conditions for validity.

4.12.3. Worked examples

Example 1. Celsius from Fahrenheit. Make C the subject of $F = \frac{9}{5}C + 32$.

$$\begin{aligned} F &= \tfrac{9}{5}C + 32 && \\ F - 32 &= \tfrac{9}{5}C && \text{(subtract 32 from both sides)} \\ \tfrac{5}{9}(F - 32) &= C && \text{(multiply both sides by } \tfrac{5}{9}\text{)} \\ C &= \tfrac{5}{9}(F - 32). && \text{(swap sides)} \end{aligned}$$

So $68°\text{F}$ is $\frac{5}{9}(36) = 20°\text{C}$. **Check:** $\frac{9}{5}(20) + 32 = 36 + 32 = 68$. Correct.

Example 2. A physics formula. The velocity after constant acceleration is $v = u + at$ (initial velocity u, acceleration a, time t). Make a the subject.

$$\begin{aligned} v &= u + at && \\ v - u &= at && \text{(subtract } u\text{)} \\ a &= \frac{v - u}{t}, \quad t \neq 0. && \text{(divide by } t\text{, swap sides)} \end{aligned}$$

That is the familiar "change in velocity over time". The condition $t \neq 0$ is not pedantry: at $t = 0$ no time has passed and the acceleration cannot be deduced.

Example 3. A trapezoid. The area of a trapezoid with parallel sides a, b and height h is $A = \frac{1}{2}h(a + b)$. Make b the subject.

$$
\begin{aligned}
A &= \tfrac{1}{2}h(a+b) && \\
2A &= h(a+b) && \text{(multiply both sides by 2: clear the fraction)} \\
\frac{2A}{h} &= a+b && \text{(divide by } h \neq 0\text{)} \\
b &= \frac{2A}{h} - a. && \text{(subtract } a\text{, swap sides)}
\end{aligned}
$$

Here we did not expand $h(a+b)$: dividing by h was cleaner. Expanding first ($2A = ha + hb$, then $hb = 2A - ha$, then $b = \frac{2A-ha}{h}$) gives the same answer in a different form. Both are correct; choose whichever path keeps the algebra short.

Example 4. The subject appears twice. Make x the subject of $y = \dfrac{x+1}{x-2}$.

This is the important case. The variable appears in two places, so no single "undo" chain will isolate it. The procedure handles it: clear the fraction, gather the x terms, factor.

$$
\begin{aligned}
y &= \frac{x+1}{x-2} && \\
y(x-2) &= x+1 && \text{(multiply by } x-2\text{; requires } x \neq 2\text{)} \\
yx - 2y &= x+1 && \text{(expand)} \\
yx - x &= 1+2y && \text{(gather } x \text{ terms left: subtract } x\text{, add } 2y\text{)} \\
x(y-1) &= 1+2y && \text{(factor out } x\text{)} \\
x &= \frac{1+2y}{y-1}, \quad y \neq 1. && \text{(divide by } y-1\text{)}
\end{aligned}
$$

Check with $x = 5$: $y = \frac{6}{3} = 2$, and the new formula gives $x = \frac{1+4}{1} = 5$. Correct. The condition $y \neq 1$ has meaning: no x makes $\frac{x+1}{x-2}$ equal to 1 (that would require $x+1 = x-2$, i.e. $1 = -2$). In the language of Chapter 11, you have just computed an *inverse function* and found a value outside its domain.

Example 5. A root. A pendulum's period is $T = 2\pi\sqrt{L/g}$. Make L the subject.

$$
\begin{aligned}
T &= 2\pi\sqrt{\frac{L}{g}} && \\
\frac{T}{2\pi} &= \sqrt{\frac{L}{g}} && \text{(divide by } 2\pi\text{)} \\
\frac{T^2}{4\pi^2} &= \frac{L}{g} && \text{(square both sides; both are non-negative, so nothing extraneous)} \\
L &= \frac{gT^2}{4\pi^2}. && \text{(multiply by } g\text{, swap sides)}
\end{aligned}
$$

With $g \approx 9.81\ \text{m/s}^2$ and a desired period of 2 seconds, $L \approx 9.81 \times 4/39.48 \approx 0.994$ m: a "seconds pendulum" is almost exactly a metre long. (That near-coincidence is not an accident; it was once proposed as the definition of the metre.)



Rearranging can silently divide by zero

Every time you divide by a letter or an expression, you are assuming it is not zero. Write the condition down ($t \neq 0, y \neq 1$). And when you take a square root to undo a square, remember $\pm$: from $A = \pi r^2$ you get $r = \sqrt{A/\pi}$ only because a radius cannot be negative; from $x^2 = k$ in general you get $x = \pm\sqrt{k}$.

4.12.4. Alternative approaches and choosing a method

- **The do/undo chain** is quickest when the variable appears *once*. Read off the order of operations applied to it, then reverse.
- **The general procedure** (clear fractions, gather, factor, divide) is needed when the variable appears *more than once*. It always works for formulas that are linear in the target variable.
- **Substitute numbers first, then solve** is fine when you only need one answer, not a reusable formula.
- **Not every formula can be rearranged.** $y = x + \sin x$ or $y = xe^x$ cannot be solved for x with elementary operations. Then numerical methods such as Newton's method take over (Volume 2, Chapter 18 , Volume 2, Chapter 32).

4.12.5. Where you'd use this

- **Electronics.** Ohm's law $V = IR$ has three useful subjects: $I = V/R$ (12 V across a 4 Ω heater draws 3 A), $R = V/I$ (which resistor limits an LED's current), and V itself.
- **Travel.** $d = vt$ rearranges to $t = d/v$: 150 miles at an average 60 mph takes 2.5 hours; and to $v = d/t$ for the average speed you need to arrive on time.
- **Spreadsheets and code.** Spreadsheets compute forwards only: you put C in a cell and F comes out. To go the other way you either rearrange the formula yourself or use a numerical "goal seek". Rearranging once, by hand, is faster, exact, and testable.

Computer-science connection: inverting a function in code

Rearranging a formula is writing the inverse function. Testing it is easy: round-trip a few values.

```python
def f_from_c(c):
    return 9/5 * c + 32

def c_from_f(f):
    return 5/9 * (f - 32)      # the rearranged formula

for c in (-40, 0, 37, 100):
    f = f_from_c(c)
    print(c, f, round(c_from_f(f), 10))
```

```
-40 -40.0 -40.0
0 32.0 0.0
37 98.60000000000001 37.0
100 212.0 100.0
```

Two things to notice. -40 is where the scales agree (solve $C = \frac{9}{5}C + 32$ to see why). And $37 \times 1.8 + 32$ printed as `98.60000000000001`, because 1.8 has no exact binary representation. The algebra is exact; the floating-point arithmetic is not. Round-trip tests in real code compare with a tolerance (Volume 2, Chapter 32).



Rearrange and verify

Pick a formula from physics or finance and a target variable. Apply operations step by step from a palette ("subtract u from both sides", "factor out x"); the lab refuses illegal moves and flags any division by an expression that could be zero. When you finish, it round-trips random values through your rearranged formula to confirm it is correct.

US curriculum

"Rearrange formulas to highlight a quantity of interest" is an **Algebra I** standard (A-CED.A.4), typically grades 8–9. Formulas with the subject appearing twice, or under roots, are common in **Algebra II** and in high-school **physics** and **chemistry**.

4.13. Translating words into algebra

4.13.1. The problem

Real problems arrive in words, not symbols. "A taxi charges a fixed fee of $3 plus $1.80 per mile. How far can you go for $30?" The mathematics here is trivial once written down; the hard part is the writing down. Translation is a skill in its own right, and it has a few traps that catch even fluent readers.

4.13.2. A phrase book

Words	Algebra
the sum of a and b; a increased by b	$a + b$
the difference of a and b; a minus b	$a - b$
b less than a; a decreased by b	$a - b$ (note the order)
the product of a and b; a times b	ab
the quotient of a and b; a per b	$\frac{a}{b}$
twice n; double n	$2n$
the square of the sum of a and b	$(a + b)^2$
the sum of the squares of a and b	$a^2 + b^2$
a is 20% more than b	$a = 1.2b$
a exceeds b by 5	$a - b = 5$
three consecutive integers	$n,\ n + 1,\ n + 2$
an even number; an odd number	$2k$; $2k + 1$ (k an integer)

Example 28. Procedure — Translating a word problem

Input: a problem stated in words.

Steps:

1. Identify the quantity asked for. Give it a letter and *write down what the letter means, with units* ("let m = distance in miles").
2. Express every other quantity in the problem in terms of that letter (and any given numbers).
3. Find the sentence that says two things are equal (or that one exceeds another) and write it as an equation.
4. Solve the equation (Chapter 5).
5. Answer the original question in words, with units, and check the answer against the original wording, not against your equation.

Output: an equation modelling the problem, and its interpreted solution.

4.13.3. Worked examples

Example 1. The taxi. Let m be the distance in miles. The cost in dollars is $3 + 1.80m$. A budget of \$30 means

$$\begin{aligned} 3 + 1.80m &= 30 \\ 1.80m &= 27 && \text{(subtract 3)} \\ m &= 15. && \text{(divide by 1.80)} \end{aligned}$$

You can travel 15 miles. **Check in words:** \$3 plus 15 miles at \$1.80 is $3 + 27 = 30$ dollars. Correct. The expression $3 + 1.80m$ is a *linear cost model*, a fixed cost plus a rate; Chapter 5 builds a whole toolkit around models of this shape.

Example 2. Consecutive integers. Three consecutive integers add to 87. Find them.

Let the smallest be n; the others are $n + 1$ and $n + 2$.

$$\begin{aligned} n + (n+1) + (n+2) &= 87 \\ 3n + 3 &= 87 && \text{(collect like terms)} \\ 3n &= 84 && \text{(subtract 3)} \\ n &= 28. && \text{(divide by 3)} \end{aligned}$$

The integers are $28, 29, 30$. **Check:** $28 + 29 + 30 = 87$.

Example 3. "Less than" reverses. "Five less than twice a number is 17." The phrase "five less than X" means $X - 5$, so the equation is $2n - 5 = 17$, giving $2n = 22$, $n = 11$. Writing $5 - 2n = 17$ is the classic mistake: English puts the 5 first, algebra puts it last.



The students-and-professors problem

"At a university there are six times as many students as professors. Write an equation using S for the number of students and P for professors." A famous 1980s study found that a large fraction of *engineering* students wrote $6S = P$. The words "six times… students" sit next to each other, so the 6 gets attached to S. But there are more students, so the professors' number must be multiplied: $S = 6P$.

The defence: *test the equation with numbers.* If there are 10 professors there are 60 students, and $S = 6P$ gives $60 = 60$; $6S = P$ gives $360 = 10$. Also make sure letters stand for *numbers of things* ("S = number of students"), not for the things themselves ("S = students").



Applications: where expressions and formulas live

Spreadsheets are the most widely used algebra system in the world: a cell formula such as `=B2*(1+C2)-D2` is an expression whose variables are other cells, and "fill down" substitutes new values for each row. Unit conversions, loan payments, dosage calculations, body-mass index, and every physics formula you will ever look up are expressions waiting for substitution. And the "think of a number" trick that opened this chapter is a tiny instance of *program verification*: proving that a computation produces a certain result for every input, by reasoning symbolically rather than testing.

4.14. Variables in mathematics and in programming

4.14.1. The problem

A programmer reading algebra brings strong habits from code, and some of them mislead. The line

```python
x = x + 1
```

is perfectly ordinary in Python, C, or Java. As a mathematical equation it is a contradiction: subtract x from both sides and you get $0 = 1$. Same symbols, entirely different meanings. Where exactly do the two notions of "variable" and "equals" part company?

4.14.2. Two meanings of "="

	Mathematics	**Most programming languages**
$=$ means	a *claim* that two things are equal (a relation, symmetric)	an *instruction*: evaluate the right, store it in the left (assignment)
`x = x + 1`	a false statement for every x ($0 = 1$)	"increment x"
Equality test	$=$	`==` (C, Python, Java), `=` in SQL `WHERE`, `===` in JavaScript
Assignment	$:=$ or "let x = ..." (definitions)	`=`, or `:=` (Pascal, Go, Python's walrus), `←` ®
A variable is	a name for a value that is fixed within the argument, even if unknown	a name for a storage location whose contents change over time
Order of lines	irrelevant to truth (all statements hold simultaneously)	crucial (each line changes state)

```python
x = 5          # assignment: bind the name x to the value 5
x = x + 1      # read x (5), add 1, rebind x to 6
print(x)       # 6
print(x == x + 1)   # equality test: is 6 equal to 7?  False
```

```
6
False
```

Within a single mathematical argument, a variable never changes value. When we wrote $2x + 3 = 11$, then $2x = 8$, then $x = 4$, we did not change x: we produced a sequence of equivalent *statements* about the same unknown. In code, by contrast, the same name can hold different values at different moments, which is why reasoning about imperative programs requires talking about *time* ("after line 3, x is 6").

Computer-science connection: closing the gap

Several corners of computing deliberately make programming variables behave like mathematical ones.

- **Functional languages** (Haskell, Clojure, Elm, and much of Scala or F#) make bindings immutable: once `x` is defined it never changes, so `x = x + 1` is either rejected or defines an infinite recursion. In exchange, you can substitute equals for equals anywhere, exactly as in algebra ("referential transparency"), which makes code easier to reason about and to parallelise.
- **Static single assignment (SSA)** form, used inside almost every optimising compiler (LLVM, GCC, V8), rewrites a program so that each variable is assigned exactly once: `x = x + 1` becomes `x2 = x1 + 1`. The compiler converts your mutable variables into mathematical ones precisely so it can apply algebraic rewrites safely.
- **Constraint and logic languages** (Prolog, SQL `WHERE`, spreadsheets, solvers such as Z3) use `=` in the mathematical sense: a relationship to be satisfied, not a command. You state $2x + 3 = 11$ and the system finds x.
- **Computer algebra systems** treat expressions as trees (Section 4.5) and implement expand, collect, factor, and solve as tree rewriting. SymPy's `expandx + 3)(x + 2` **returns** `x *2 + 5*x + 6` by exactly the distributive-law steps you did by hand. Learning those steps is still worthwhile: you cannot sanity-check a CAS's output, choose a useful form, or notice when it has divided by something that might be zero, unless you can do a small case yourself.



Bound and free variables

A variable is **free** in an expression if the expression's value depends on it, and **bound** if it is only a placeholder inside some construction. In $\sum_{i=1}^{n} i^2$ (the sum of i^2 as i runs from 1 to n, see Chapter 14), n is free and i is bound: renaming i to k changes nothing. Programmers know this as *scope*: a loop counter is local to the loop, exactly like a bound variable.

US curriculum

Translating verbal descriptions into expressions and equations runs from **grade 6** through **Algebra I**, and "modelling" is a cross-cutting theme in the Common Core high-school standards. The programming connections are not part of the standard math curriculum but appear in **AP Computer Science A/Principles**, where students must distinguish `=` from `==`.

4.15. Exercises: equations, rearranging, translating

1. Which of these operations, applied to both sides of an equation, always gives an equivalent equation? (a) add 7; (b) multiply by x; (c) divide by 3; (d) square; (e) subtract $2x$.
2. Make r the subject of $C = 2\pi r$.
3. Make s the subject of $v^2 = u^2 + 2as$.
4. Make w the subject of $P = 2(\ell + w)$.
5. The lens formula is $\dfrac{1}{f} = \dfrac{1}{u} + \dfrac{1}{v}$. Make v the subject.
6. Make x the subject of $y = \dfrac{2x + 3}{x - 1}$. For which value of y does your formula fail, and why?
7. Translate and solve: "Five less than twice a number is 17."
8. A taxi charges \$3 plus \$1.80 per mile. Write an expression for the cost of an m-mile trip and find the cost of a 12-mile trip.
9. A rectangle's length is 3 cm more than twice its width, and its perimeter is 54 cm. Find its dimensions.
10. A programmer claims that `total = total + price` "proves" $\text{price} = 0$. Explain the confusion in one or two sentences.

▼ **Answers**

1. (a), (c), (e) always give equivalent equations. (b) fails if $x = 0$ (it can introduce the solution $x = 0$). (d) can introduce extraneous solutions ($x = 3$ becomes $x^2 = 9$, which also allows -3).
2. $r = \dfrac{C}{2\pi}$.
3. $v^2 - u^2 = 2as$, so $s = \dfrac{v^2 - u^2}{2a}$, with $a \neq 0$.

4. $\frac{P}{2} = \ell + w$, so $w = \frac{P}{2} - \ell$ (equivalently $\frac{P-2\ell}{2}$).
5. $\frac{1}{v} = \frac{1}{f} - \frac{1}{u} = \frac{u-f}{fu}$, so $v = \dfrac{fu}{u-f}$, with $u \neq f$. (Check: $f = 10, u = 30$ gives $v = 300/20 = 15$, and $\frac{1}{30} + \frac{1}{15} = \frac{3}{30} = \frac{1}{10}$. Correct.)
6. $y(x-1) = 2x + 3 \Rightarrow yx - y = 2x + 3 \Rightarrow yx - 2x = y + 3 \Rightarrow x(y-2) = y + 3 \Rightarrow x = \dfrac{y+3}{y-2}$. It fails at $y = 2$: no x gives $y = 2$, since $2x + 3 = 2(x-1)$ would mean $3 = -2$.
7. $2n - 5 = 17 \Rightarrow 2n = 22 \Rightarrow n = 11$. Check: $2(11) - 5 = 17$.
8. $C = 3 + 1.80m$. For $m = 12$: $3 + 21.60 = 24.60$ dollars.
9. Let width w cm, length $2w + 3$ cm. $2\big(w + (2w + 3)\big) = 54 \Rightarrow 6w + 6 = 54 \Rightarrow w = 8$. Length $2(8) + 3 = 19$ cm. Check: $2(8 + 19) = 54$.
10. In code, `=` is assignment: it stores the old total plus the price as the new total. Two different moments in time are involved, so there is no equation to subtract from. Read mathematically as $t = t + p$, it would indeed force $p = 0$, but that is not what the code says.

4.16. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Grammar and parser playground

4.17. Mastery check

1. Which of these are like terms: $3x^2y$, $-2xy^2$, $5yx^2$, $7xy$?
2. Evaluate $(a-b)^2 - (a^2 - b^2)$ when $a = 3, b = -1$.
3. Expand and simplify $2(x+3)(x-1) - (x-2)^2$.
4. Factor completely: $18p^3q - 27p^2q^2$.
5. The surface area of a closed cylinder is $S = 2\pi r^2 + 2\pi rh$. Make h the subject.
6. Make x the subject of $a(x-b) = c(x+d)$, and state any condition.
7. Is $4(x-2) - 2(2x-4) = 0$ a conditional equation, an identity, or a contradiction?
8. Translate into an equation: "A number's square exceeds the number itself by 12." Verify that 4 and -3 both satisfy it.

Decision rule: All correct without hesitation → move on to Chapter 5. Stuck on 1–2 → reread Section 4.3, Section 4.7, or Section 4.12 as appropriate. Stuck on more → work the three exercise sets in this chapter first, then retry.

▼ **Mastery check answers**

1. $3x^2y$ and $5yx^2$ (same variable part, x^2y). $-2xy^2$ has y squared instead, and $7xy$ has lower powers.
2. $(3-(-1))^2-(3^2-(-1)^2)=4^2-(9-1)=16-8=8$.
3. $(x+3)(x-1)=x^2+2x-3$, so the first part is $2x^2+4x-6$. $(x-2)^2=x^2-4x+4$. Subtracting (negate every term): $2x^2+4x-6-x^2+4x-4=x^2+8x-10$.
4. GCD of 18 and 27 is 9; lowest powers p^2, q. $9p^2q(2p-3q)$.
5. $S-2\pi r^2=2\pi rh$, so $h=\dfrac{S-2\pi r^2}{2\pi r}=\dfrac{S}{2\pi r}-r$ (with $r>0$).
6. $ax-ab=cx+cd \Rightarrow ax-cx=ab+cd \Rightarrow x(a-c)=ab+cd \Rightarrow x=\dfrac{ab+cd}{a-c}$, provided $a\neq c$.
7. Identity: $4x-8-4x+8=0$, i.e. $0=0$, true for every x.
8. $n^2=n+12$ (or $n^2-n=12$). $n=4$: $16=16$. $n=-3$: $9=-3+12=9$. Solving such equations systematically is the subject of Chapter 8.

Chapter 5. Linear Equations and Inequalities

A taxi charges a fixed fee plus a rate per mile. How far can you go for $25? When does one phone plan become cheaper than another? Where should an animation be 40% of the way through its move? Each is a linear question, and each is answered with the same small set of reversible moves. Linear equations are probably the most-used mathematics there is: every unit conversion, budget, dosage, "how long until?" and "how many will fit?" is one. This chapter turns those moves into an algorithm, then extends it from equations (one answer) to inequalities (a whole range of answers), absolute values (distances), and finally to the straight line itself, the picture behind all of it.

Where this chapter sits

You need: the vocabulary of expressions, terms, coefficients and the distributive law from Chapter 4; signed-number arithmetic from Chapter 1; fraction arithmetic and lowest common denominators from Chapter 2.

This chapter unlocks: systems of equations (Chapter 6), where several linear equations must hold at once; coordinate geometry (Chapter 10); the general idea of a function (Chapter 11); and, much later, the "local linearity" at the heart of calculus (Volume 2, Chapter 17) and the whole of linear algebra (Volume 2, Chapter 26 , Volume 2, Chapter 27).

Fast path: if the mastery check at the end of the chapter is easy, skip to Chapter 6.

5.1. What makes an equation linear?

5.1.1. The problem

A taxi charges $3.00 to start the meter plus $2.50 per mile. Your ride cost $23.00. How far did you travel?

Call the distance d miles. Then the cost is $3 + 2.5d$, and the information you have is

$$3 + 2.5d = 23.$$

You can probably see the answer quickly: take off the $3 start fee to leave $20, and $20 at $2.50 a mile is 8 miles. That two-step mental calculation is exactly the algorithm this chapter formalises. Before we do that, let us name what kind of equation this is, because the name tells you which tools apply.

5.1.2. Vocabulary

Linear equation

A **linear equation** in one unknown x is an equation that can be rearranged into the form

$$ax + b = 0, \qquad a \neq 0,$$

where a and b are constants. The unknown appears only to the **first power**: no x^2, no $\sqrt{x}$, no $1/x$, no x multiplied by another unknown. The name comes from the Latin *linea*, "line": graph $y = ax + b$ and you get a straight line (Section 5.10).

Solution, root, solution set

A **solution** of an equation is a value of the unknown that makes the equation a true statement. For equations of the form $f(x) = 0$, a solution is also called a **root**. The **solution set** is the collection of all solutions. For a linear equation it contains exactly one number, no numbers, or every number (Section 5.4). For an inequality it is usually a whole interval.

Equivalent equations

Two equations are **equivalent** if they have exactly the same solution set. $2x + 1 = 7$, $2x = 6$ and $x = 3$ are equivalent. **Solving** an equation means producing a chain of equivalent equations that ends in something so simple, like $x = 3$, that the solution set can be read off.

Which of these are linear in x?

- $5x - 7 = 2x + 8$ — yes. Collect terms and it becomes $3x - 15 = 0$.
- $\frac{1}{2}(x + 4) = \sqrt{2}\,x$ — yes. The coefficient $\sqrt{2}$ is irrational, but it is still a constant multiplying the first power of x.
- $x^2 = 9$ — no, the unknown is squared (that is a quadratic; see Chapter 8).
- $\frac{3}{x} = 6$ — not as written, because x is in a denominator. Multiplying by x turns it into $3 = 6x$, which is linear, but only after you have noted that $x \neq 0$. We will come back to this subtlety.
- $xy = 6$ — no; it is a product of two unknowns.

US curriculum

One-step and two-step equations appear in grades 6–7 (**Pre-Algebra**, Common Core 7.EE). Multi-step linear equations, literal equations, linear inequalities and absolute-value equations are the backbone of **Algebra I** (grades 8–9, standards A-REI.3 and A-CED.1). Compound and absolute-value inequalities are often revisited in **Algebra II**.

5.2. Solving linear equations in one unknown

5.2.1. The problem

"I think of a number, multiply it by 3, subtract 5, and get 16. What was my number?"

Written as an equation, $3x - 5 = 16$. The puzzle describes a little **pipeline** of operations applied to the unknown:

$$x \xrightarrow{\times 3} 3x \xrightarrow{-5} 3x - 5 = 16.$$

To recover x you run the pipeline backwards, undoing each stage with its inverse, last stage first:

$$16 \xrightarrow{+5} 21 \xrightarrow{\div 3} 7.$$

The number was 7. Check: $3 \cdot 7 - 5 = 16$. That is the entire idea of solving a linear equation. Everything else is bookkeeping for when the pipeline is messier: when x appears on both sides, inside brackets, or inside fractions.

5.2.2. Intuition: undo in reverse order

You put on socks, then shoes. To undo, you take off shoes first, then socks. Inverse operations undo in reverse order for the same reason: the last thing done to x is the outermost layer, so it has to come off first. In $3x - 5$, the subtraction of 5 happened *after* the multiplication by 3 (order of operations), so it is undone *first*.



Inverse operation

An **inverse operation** undoes another. Adding k and subtracting k are inverses; multiplying by k and dividing by k are inverses, **provided** $k \neq 0$. Applying an operation and then its inverse returns you to where you started.

The second intuition is the **balance** from Chapter 4: an equation is a pair of scales in equilibrium. Whatever you do to one pan, you must do to the other, or the balance tips and the statement is no longer the one you started with.

5.2.3. Why it works

Why is it legitimate to "do the same thing to both sides"? Suppose $x = s$ is a solution of $L = R$, meaning that when x is replaced by s the two sides are the same number. Adding 5 to two equal numbers gives two equal numbers; multiplying two equal numbers by 3 gives two equal numbers. So every solution of the old equation is a solution of the new one.

That is only half of what we need. We also need the reverse: the new equation should not have *gained* solutions. That is where the inverse matters. If you added 5, you can subtract 5 to get back exactly the original equation, so any solution of the new equation is also a solution of the old one. Each step is **reversible**, and reversible steps produce **equivalent** equations.

Now you can see why multiplying both sides by zero is forbidden. Multiplying $2x = 6$ by 0 gives $0 = 0$, which is true for every x. Information has been destroyed and cannot be recovered, because there is no "divide by 0" to undo it. The two legal moves, stated precisely:

$$\begin{aligned} L = R &\iff L + k = R + k && \text{for any number or expression } k, \\ L = R &\iff kL = kR && \text{for any } k \neq 0. \end{aligned}$$

The double arrow $\iff$ means "if and only if": each equation implies the other, so their solution sets are identical. Subtraction is adding $-k$; division is multiplying by $1/k$. Simplifying a side (expanding brackets, collecting like terms, as in Chapter 4) is the third legal move, since it does not change the value of that side at all.

5.2.4. The method

Example 29. Procedure — solve a linear equation in one unknown

Input: an equation in which the unknown x appears only to the first power.

Steps:

1. **Clear fractions:** if any term has a numeric denominator, multiply every term on both sides by the lowest common denominator (Section 5.3).
2. **Expand:** remove brackets with the distributive law, watching signs.
3. **Simplify each side:** collect like terms separately on the left and on the right.
4. **Collect:** add or subtract so that all x-terms are on one side and all constant terms on the other. (Moving the x-terms toward the side with the larger x-coefficient keeps the coefficient positive.)
5. **Isolate:** you now have $ax = c$. If $a \neq 0$, divide both sides by a: $x = c/a$. If $a = 0$, go to Section 5.4: there is no solution or every number is a solution.
6. **Check:** substitute the answer into the *original* equation and evaluate each side separately.

Output: the solution set: one number, the empty set, or all real numbers.

5.2.5. Worked examples

Example 1. Solve $3(2x - 5) = 4x + 7$.

$$\begin{aligned} 3(2x - 5) &= 4x + 7 && \\ 6x - 15 &= 4x + 7 && \text{(expand: } 3 \cdot 2x = 6x,\ 3 \cdot (-5) = -15\text{)} \\ 6x - 4x - 15 &= 4x - 4x + 7 && \text{(subtract } 4x \text{ from both sides)} \\ 2x - 15 &= 7 && \text{(collect like terms)} \\ 2x - 15 + 15 &= 7 + 15 && \text{(add 15 to both sides)} \\ 2x &= 22 && \\ x &= 11 && \text{(divide both sides by 2)} \end{aligned}$$

Check: left side $3(2 \cdot 11 - 5) = 3 \cdot 17 = 51$; right side $4 \cdot 11 + 7 = 51$. Equal, so $x = 11$.

Example 2. Solve $5 - 2(x - 3) = 3x - 4$.

The danger here is the minus sign in front of the bracket. $-2(x - 3)$ means -2 multiplied by *each* term inside: $(-2)(x) + (-2)(-3) = -2x + 6$.

$$
\begin{aligned}
5 - 2(x - 3) &= 3x - 4 && \\
5 - 2x + 6 &= 3x - 4 && \text{(expand; } (-2)(-3) = +6\text{)} \\
11 - 2x &= 3x - 4 && \text{(collect constants on the left)} \\
11 - 2x + 2x &= 3x + 2x - 4 && \text{(add } 2x\text{; the right side has the larger } x\text{-coefficient)} \\
11 &= 5x - 4 && \\
15 &= 5x && \text{(add 4 to both sides)} \\
3 &= x && \text{(divide by 5)}
\end{aligned}
$$

Check: left $5 - 2(3 - 3) = 5 - 0 = 5$; right $3 \cdot 3 - 4 = 5$. Equal.

Notice the choice made at the fourth line. Adding $2x$ to both sides moved the x-terms to the right, where they ended up with the positive coefficient 5. Subtracting $3x$ instead would have given $11 - 5x = -4$, then $-5x = -15$, then $x = 3$: the same answer, but with two extra negative signs to get wrong. An equation with x on the right, like $3 = x$, is perfectly fine; equality is symmetric.

5.2.6. Alternative approaches

- **Reverse flowchart.** For an equation where x appears exactly once, such as $\frac{4x+1}{3} - 2 = 5$, write the pipeline ($\times 4$, $+1$, $\div 3$, -2) and run it backwards from 5: $5 + 2 = 7$, $7 \times 3 = 21$, $21 - 1 = 20$, $20 \div 4 = 5$. So $x = 5$. Fast and transparent, but useless once x appears more than once.
- **Guess and check (trial and improvement).** Try values, see whether the left side is too big or too small, adjust. Fine for building intuition, slow and imprecise otherwise. It is, however, exactly what numerical methods do for equations that *cannot* be solved algebraically (Volume 2, Chapter 32).
- **Graphing.** Plot $y = 3(2x - 5)$ and $y = 4x + 7$ and read off where they cross. This gives understanding and a sanity check, but only approximate answers by eye. We return to this view in Section 5.10.
- **Rearrange to $ax + b = 0$ and use $x = -b/a$.** This is what a computer does (see the code below). By hand it is no faster than the procedure.

5.2.7. Choosing a method

For one-off equations by hand, use the procedure: it never fails and every step is checkable. Use the reverse flowchart when x appears once; it is quicker and harder to get wrong. Use a graph when you want to *see* the solution or to check a suspicious answer. Use guess-and-check only when the equation is not linear and no algebraic method is available.



Distributing a minus sign, and other classic slips

- $-2(x-3)$ is $-2x+6$, not $-2x-6$. The sign multiplies every term in the bracket.
- $2(x+3)=10$ does not become $2x+3=10$. Multiply *every* term inside the bracket, or divide both sides by 2 first: $x+3=5$.
- "Moving a term to the other side changes its sign" is a shortcut, not a rule. It is really "add the opposite to both sides". If you forget why it works, you will eventually apply it to a factor (moving a 3 that multiplies x across as -3), which is wrong; a multiplying factor moves across as a division.
- Dividing both sides by an *expression* that might be zero can silently lose a solution. Dividing $x(x-2)=3x$ by x loses the solution $x=0$. (That equation is quadratic, not linear, but the habit starts here.)
- Check in the *original* equation. Checking in an intermediate line only confirms the steps after the mistake.

5.2.8. Where you'd use this

Any time a formula gives you the answer to the wrong question, you solve a linear equation to turn it round.

- **Shopping.** A receipt says you paid \$54.00 including 8% sales tax. The pre-tax price p satisfies $1.08p=54$, so $p=54/1.08=50$. (Subtracting 8% of \$54 instead gives \$49.68, a classic slip: the 8% was of the price, not of the total.)
- **Chemistry and medicine.** Diluting a stock solution keeps the amount of solute fixed: $C_1V_1=C_2V_2$. To make 500 mL of 0.1 M solution from 2 M stock, solve $2V_1=0.1\times 500$: $V_1=25$ mL of stock, topped up to 500 mL. Nurses do the same calculation for drug dilutions.
- **Mechanics.** The constant-acceleration equation $v=u+at$ is linear in each letter. A car accelerating from rest at $3\ \text{m/s}^2$ reaches $27\ \text{m/s}$ (about 60 mph) when $27=0+3t$, i.e. after $t=9$ s.
- **Computing.** A drawing program maps a world coordinate x to a screen pixel by $p=40+8x$ (a 40-pixel margin, 8 pixels per unit). When the user clicks pixel 360, the program solves $40+8x=360$ to find that they clicked world coordinate $x=40$. Every zoomable map, chart and game camera inverts a linear map like this on every mouse event.

Computer-science connection: solving is inverting a pipeline

A linear expression is a composition of invertible functions, and solving is applying their inverses in reverse order, exactly like popping an undo stack or unwinding a sequence of transformations in a graphics pipeline. Once both sides are simplified, every linear equation reduces to $(a - c)x = d - b$, so a computer needs no step-by-step reasoning at all. Using Python's `Fraction` type keeps the arithmetic exact, so $\frac{8}{3}$ is not printed as 2.6666666666666665:

```python
from fractions import Fraction

def solve_linear(a, b, c, d):
    """Solve a*x + b = c*x + d exactly.

    Returns a Fraction, the string 'none', or the string 'all'.
    """
    a, b, c, d = map(Fraction, (a, b, c, d))
    coeff = a - c          # collect x-terms on the left
    const = d - b          # collect constants on the right
    if coeff == 0:
        return "all" if const == 0 else "none"
    return const / coeff   # undo the multiplication

print(solve_linear(5, -7, 2, 8))      # 5x - 7 = 2x + 8
print(solve_linear(3, 1, 3, 4))       # 3x + 1 = 3x + 4
print(solve_linear(2, 6, 2, 6))       # 2x + 6 = 2x + 6
print(solve_linear("1/2", "1/3", "1/4", 1))
```

```
5
none
all
8/3
```

The hard part, as always, is not the arithmetic but getting the equation into that shape: parsing, expanding and collecting. That is what a computer algebra system does (Chapter 4).



> *Restoring and balancing*
>
> The word *algebra* comes from the title of al-Khwārizmī's ninth-century book *al-Kitāb al-mukhtaṣar fī ḥisāb al-jabr wa-l-muqābala*. *Al-jabr* ("restoration") was the move of adding a term to both sides to cancel a subtracted quantity; *al-muqābala* ("balancing") was cancelling like terms on opposite sides. Those are precisely steps 4 and 3 of our procedure. Al-Khwārizmī's own name, Latinised, gave us the word *algorithm*. Solving linear equations is where both words were born.



Balance-scale solver

Show an equation as a balance scale with x-boxes and unit weights on each pan. The learner chooses an operation ("add 3 to both", "divide both by 2"); the scale applies it to both pans and the symbolic equation updates beside it. Illegal moves, such as acting on one pan only or dividing by zero, visibly tip the scale. A counter compares the learner's number of steps with the minimum.

5.3. Equations with fractions and decimals

5.3.1. The problem

Problems about shares and proportions ("a third of one amount, less a quarter of another") arrive full of fractions. A typical result is

$$\frac{x+1}{3} - \frac{x-2}{4} = 1.$$

You could work with the fractions throughout, but there is a better move. Every term can be multiplied by the same nonzero number without changing the solution set, so multiply by a number that cancels *every* denominator at once.



Lowest common denominator (LCD)

The **lowest common denominator** of several fractions is the least common multiple of their denominators: the smallest positive number each denominator divides into. For $\frac{1}{3}$ and $\frac{1}{4}$ it is 12; for $\frac{1}{4}$ and $\frac{1}{6}$ it is 12, not 24. (See Chapter 2.) Multiplying an equation through by the LCD is called **clearing fractions**.

5.3.2. Worked example

Solve $\frac{x+1}{3} - \frac{x-2}{4} = 1$. The LCD of 3 and 4 is 12.

$$\begin{aligned}
12 \cdot \frac{x+1}{3} - 12 \cdot \frac{x-2}{4} &= 12 \cdot 1 && \text{(multiply every term by 12)} \\
4(x+1) - 3(x-2) &= 12 && \text{(12/3 = 4, 12/4 = 3)} \\
4x + 4 - 3x + 6 &= 12 && \text{(expand; note } -3 \cdot (-2) = +6\text{)} \\
x + 10 &= 12 && \text{(collect like terms)} \\
x &= 2 && \text{(subtract 10)}
\end{aligned}$$

Check: $\frac{2+1}{3} - \frac{2-2}{4} = 1 - 0 = 1$. Correct.

The single most common error here is in line 2. The fraction bar is a grouping symbol: $\frac{x-2}{4}$ means $\frac{1}{4}(x-2)$, so after clearing, the -3 multiplies the whole of $(x-2)$. Writing the brackets explicitly at this stage, before expanding, prevents the mistake.

Decimals are fractions in disguise, cleared by a power of 10. For $0.05x + 0.12(200 - x) = 17$, multiply through by 100:

$$\begin{aligned} 5x + 12(200 - x) &= 1700 && \text{(multiply every term by 100)} \\ 5x + 2400 - 12x &= 1700 && \text{(expand)} \\ -7x &= -700 && \text{(collect; subtract 2400)} \\ x &= 100 && \text{(divide by } -7\text{)} \end{aligned}$$

Check: $0.05(100) + 0.12(100) = 5 + 12 = 17$.

5.3.3. When the unknown is in a denominator

Strictly, $\frac{2}{x} + 3 = \frac{5}{x}$ is not linear, but it *becomes* linear after multiplying by x. The catch is that x could be zero, and multiplying by zero is not reversible. So first record the restriction $x \neq 0$, then proceed:

$$\begin{aligned} x \cdot \frac{2}{x} + 3x &= x \cdot \frac{5}{x} \qquad (x \neq 0) \\ 2 + 3x &= 5 \\ x &= 1. \end{aligned}$$

$x = 1$ satisfies the restriction, and the check $2 + 3 = 5$ confirms it.



Extraneous solutions

Multiplying by an expression that can be zero may *introduce* a false solution. Solve $\frac{x}{x-3} = \frac{3}{x-3} + 2$. The restriction is $x \neq 3$. Multiplying by $(x - 3)$:

$$x = 3 + 2(x - 3) = 2x - 3 \quad \Longrightarrow \quad x = 3.$$

But $x = 3$ is forbidden: both fractions have zero denominators there. The original equation has **no solution**; the "3" is an **extraneous solution** created by the non-reversible step. This is why the check step in the procedure is not optional. Rational equations get full treatment alongside rational expressions in Chapter 7.

5.4. One solution, none, or infinitely many

5.4.1. The problem

Try to solve $3x + 1 = 3x + 4$. Subtract $3x$ from both sides and you get $1 = 4$. The x has vanished and left a false statement. What went wrong?

Nothing went wrong. The equation asks "which numbers x make $3x + 1$ equal to $3x + 4$?" and the answer is "none": whatever x is, the right side is always 3 bigger. The algebra correctly reported that.

5.4.2. Why there are exactly three cases

After expanding and collecting, *every* linear-looking equation $ax + b = cx + d$ becomes

$$(a - c)\,x = d - b.$$

Call the coefficient $A = a - c$ and the right side $B = d - b$. Then:

Case	What happens	Solution set
$A \neq 0$	divide by A	exactly one solution, $x = B/A$
$A = 0,\ B \neq 0$	$0 = B$ is false for every x	no solution (the empty set $\varnothing$)
$A = 0,\ B = 0$	$0 = 0$ is true for every x	every real number, $\mathbb{R}$



Identity, contradiction, conditional equation

An **identity** is an equation true for every value of the variable, such as $2(x+3) = 2x+6$. A **contradiction** is true for no value, such as $x+1 = x+2$. A **conditional equation** is true for some values and false for others, such as $2x = 6$. The symbol $\varnothing$ denotes the **empty set**, the set with no elements; $\mathbb{R}$ denotes the set of all real numbers.

Example. Solve $6x - 2(3x+1) = -2$.

$$\begin{aligned} 6x - 6x - 2 &= -2 && \text{(expand)} \\ -2 &= -2 && \text{(collect: the } x\text{-terms cancel)} \end{aligned}$$

This is always true, so the solution set is $\mathbb{R}$: every real number works. Test any value you like; $x = 100$ gives $600 - 602 = -2$.

The picture behind these cases is simple and will matter a great deal in Chapter 6. Graph each side as a line, $y = ax + b$ and $y = cx + d$. The x-coefficients a and c are the slopes. Different slopes: the lines cross exactly once. Equal slopes, different intercepts: parallel lines that never meet. Equal slopes *and* equal intercepts: the same line, meeting everywhere.



Reading 0 = 0 as x = 0

When the variable cancels and leaves $0 = 0$, the answer is "all real numbers", not "$x = 0$". When it leaves something like $0 = 3$, the answer is "no solution", not "$x = 3$". Write the solution set explicitly: $\mathbb{R}$ or $\varnothing$.

5.5. Exercises: solving equations

1. Which of these are linear equations in x? (a) $3x - 7 = 2$ (b) $x^2 = 4$ (c) $0.5(x+2) = \sqrt{2}\,x$ (d) $xy = 6$ (e) $\frac{x}{4} = 9$.
2. Solve $4x + 9 = 1$.
3. Solve $7 - (2x - 5) = 3(x+1) - 1$.
4. Solve $\dfrac{2x-1}{5} = \dfrac{x+4}{3}$.
5. Solve $\dfrac{x}{2} - \dfrac{x}{3} = 4$.

6. Classify each as identity or contradiction and give its solution set: (a) $4(x-1) = 4x - 4$ (b) $2x + 3 = 2(x + 3)$.
7. For which value of k does $kx + 2 = 3x - 1$ have no solution? Is there a value of k for which it has infinitely many?
8. A student writes: $2(x + 3) = 10 \Rightarrow 2x + 3 = 10 \Rightarrow x = 3.5$. Find the error and give the correct solution.

▼ **Answers**

1. (a), (c) and (e) are linear. (b) has x^2; (d) is a product of two unknowns.
2. $4x = 1 - 9 = -8$, so $x = -2$. Check: $-8 + 9 = 1$.
3. Expand: $7 - 2x + 5 = 3x + 3 - 1$, so $12 - 2x = 3x + 2$. Add $2x$ and subtract 2: $10 = 5x$, $x = 2$. Check: left $7 - (-1) = 8$, right $3 \cdot 3 - 1 = 8$.
4. LCD 15: $3(2x - 1) = 5(x + 4)$, so $6x - 3 = 5x + 20$, $x = 23$. Check: $45/5 = 9$ and $27/3 = 9$.
5. LCD 6: $3x - 2x = 24$, so $x = 24$. Check: $12 - 8 = 4$.
6. (a) Expanding gives $4x - 4 = 4x - 4$, then $0 = 0$: identity, solution set $\mathbb{R}$. (b) $2x + 3 = 2x + 6$ gives $3 = 6$: contradiction, solution set $\varnothing$.
7. Collect: $(k - 3)x = -3$. If $k = 3$ this reads $0 = -3$: no solution. Infinitely many would need both $k - 3 = 0$ and $-3 = 0$, which is impossible, so no such k. For every other k the unique solution is $x = -3/(k - 3)$.
8. The 2 was not multiplied into the 3: $2(x + 3) = 2x + 6$. Correctly, $2x + 6 = 10$, $2x = 4$, $x = 2$. Check: $2(5) = 10$.

5.6. Linear inequalities

5.6.1. The problem

Back in the taxi: you have \$25. How far can you go? Now the question is not "which distance costs *exactly* \$25" but "which distances cost *at most* \$25":

$$3 + 2.5d \le 25.$$

The answer is not a single number but a whole range of numbers. Inequalities are how mathematics talks about budgets, limits, tolerances, capacities and thresholds: anything with "at most", "at least", "no more than", "exceeds".

Inequality symbols

$a < b$	a is **less than** b: a lies to the left of b on the number line.
$a > b$	a is **greater than** b.
$a \leq b$	a is **less than or equal to** b ("at most").
$a \geq b$	a is **greater than or equal to** b ("at least").
$a \neq b$	a is **not equal to** b.

$<$ and $>$ are **strict** inequalities; $\leq$ and $\geq$ are **non-strict** (or **weak**). The narrow end of the symbol points at the smaller number.

Solving the taxi inequality looks exactly like solving the equation:

$$\begin{aligned} 3 + 2.5d &\leq 25 \\ 2.5d &\leq 22 \qquad \text{(subtract 3)} \\ d &\leq 8.8 \qquad \text{(divide by 2.5)} \end{aligned}$$

Any distance up to 8.8 miles (and, since distance cannot be negative, at least 0) is affordable. So far, inequalities behave like equations. There is exactly one twist.

5.6.2. Intuition: why a negative multiplier flips the sign

Start with a true statement, $2 < 4$, and multiply both sides by -1. You get -2 and -4. Which is bigger? On the number line -2 is to the right of -4, so $-2 > -4$. The inequality has reversed.

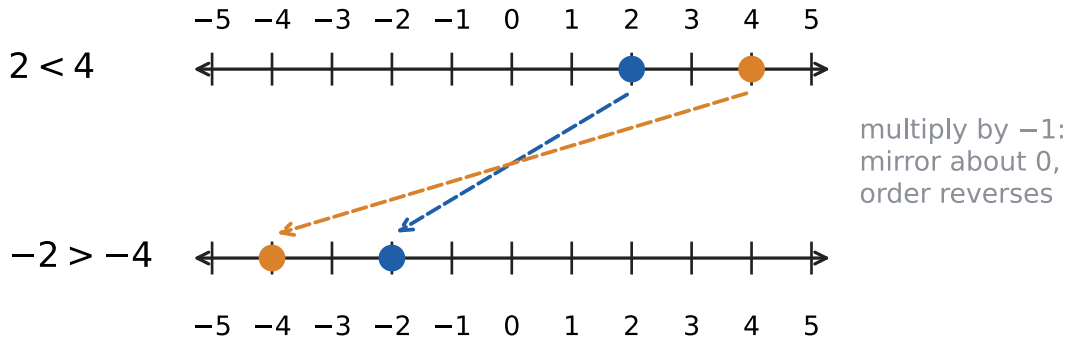


Figure 31. Multiplying by -1 reflects the number line about 0, which reverses the order of every pair of points.

Geometrically, multiplying by -1 is a mirror reflection of the number line about 0. A reflection turns left-to-right into right-to-left, so "is to the left of" becomes "is to the right of". Multiplying by any negative number is a reflection followed by a stretch, and the stretch does not change order, so every negative multiplier reverses the inequality.

5.6.3. Why it works

Here is the argument in symbols; it is short and it shows exactly where the sign of the multiplier enters. Recall that $a < b$ means the same as $b - a > 0$ (the gap from a up to b is positive).

Suppose $a < b$, so $b - a > 0$, and let k be any number. Then

$$kb - ka = k(b - a).$$

- If $k > 0$: a positive times a positive is positive, so $kb - ka > 0$, that is, $ka < kb$. The order is kept.
- If $k < 0$: a negative times a positive is negative, so $kb - ka < 0$, that is, $ka > kb$. The order is reversed.
- If $k = 0$: both sides become 0 and all information is lost, just as with equations.

Adding the same number to both sides never changes the order, because $(b + k) - (a + k) = b - a$: the gap is unchanged. So the rules for inequalities are the rules for equations, plus one exception.

$$\begin{aligned} a < b &\iff a + k < b + k && \text{for any } k, \\ a < b &\iff ka < kb && \text{for } k > 0, \\ a < b &\iff ka > kb && \text{for } k < 0 \quad \textbf{(reverse the sign).} \end{aligned}$$

The same rules hold with $\leq$ in place of $<$. Swapping the two sides also reverses the symbol: $3 < x$ says the same thing as $x > 3$.

5.6.4. The method

Example 30. Procedure — solve a linear inequality

Input: an inequality, using $<$, $\leq$, $>$ or $\geq$, in which x appears only to the first power.

Steps:

1. Clear fractions by multiplying by the (positive) LCD. Positive, so no flip.
2. Expand brackets and collect like terms on each side.
3. Add or subtract to put x-terms on one side and constants on the other.
4. Divide by the coefficient of x. **If that coefficient is negative, reverse the inequality symbol.**
5. Write the solution set (as an inequality, in interval notation, and on a number line).
6. Check: pick one test value inside the solution set and one outside, and substitute both into the original inequality.

Output: the solution set, usually an interval.

5.6.5. Worked example

Solve $7 - 3x \leq 19$.

$$
\begin{aligned}
7 - 3x &\leq 19 \\
-3x &\leq 12 && \text{(subtract 7)} \\
x &\geq -4 && \text{(divide by } -3\text{: reverse } \leq \text{ to } \geq\text{)}
\end{aligned}
$$

Check with a value inside, $x = 0$: $7 - 0 = 7 \leq 19$ is true. A value outside, $x = -5$: $7 + 15 = 22 \leq 19$ is false. The boundary, $x = -4$: $7 + 12 = 19 \leq 19$ is true, so the endpoint belongs to the solution set, as $\geq$ says. Solution set: $[-4, \infty)$.

5.6.6. Alternative approach: avoid the flip altogether

You can sidestep the sign reversal by moving the x-term to whichever side makes its coefficient positive:

$$
\begin{aligned}
7 - 3x &\leq 19 \\
7 &\leq 19 + 3x && \text{(add } 3x\text{)} \\
-12 &\leq 3x && \text{(subtract 19)} \\
-4 &\leq x && \text{(divide by } +3\text{: no flip)}
\end{aligned}
$$

$-4 \leq x$ is the same statement as $x \geq -4$. Many people find this method more reliable, because it replaces a rule you have to remember with a move you cannot get wrong. Use whichever you prefer; but know both, because when the coefficient is a symbol a whose sign you do not know, you need to split into cases $a > 0$ and $a < 0$ and the flip rule is unavoidable.



Inequality pitfalls

- **Dividing by a negative and forgetting to flip.** The test-value check catches this every time.
- **Flipping when dividing by a positive.** $-2 < 4x$ becomes $-\frac{1}{2} < x$; the *number* on the other side being negative is irrelevant. Only the sign of the multiplier matters.
- **Multiplying by an expression of unknown sign.** You cannot multiply $\frac{1}{x} < 2$ by x without knowing whether x is positive or negative. Split into cases.
- **Taking reciprocals.** For two positive numbers, $a < b$ implies $\frac{1}{a} > \frac{1}{b}$ ($2 < 4$ but $\frac{1}{2} > \frac{1}{4}$). Reciprocation also reverses order.
- **Ignoring context.** $d \leq 8.8$ includes negative distances mathematically; the real answer is $0 \leq d \leq 8.8$.

5.6.7. Interval notation and set notation

A solution like "all x with $-2 \leq x < 3$" is common enough to deserve a compact notation.

Interval notation

An **interval** is an unbroken stretch of the number line. A **square bracket** means the endpoint is included; a **round bracket** (parenthesis) means it is excluded.

Inequality	Interval	Name
$a \le x \le b$	$[a, b]$	closed interval
$a < x < b$	(a, b)	open interval
$a \le x < b$	$[a, b)$	half-open interval
$a < x \le b$	$(a, b]$	half-open interval
$x \ge a$	$[a, \infty)$	unbounded (ray)
$x < b$	$(-\infty, b)$	unbounded (ray)
all real numbers	$(-\infty, \infty) = \mathbb{R}$	

The symbol ∞ (infinity) is not a number; $[a, \infty)$ means "no upper limit". Infinity therefore always gets a round bracket. On a number line, a filled dot marks an included endpoint and an open circle an excluded one.

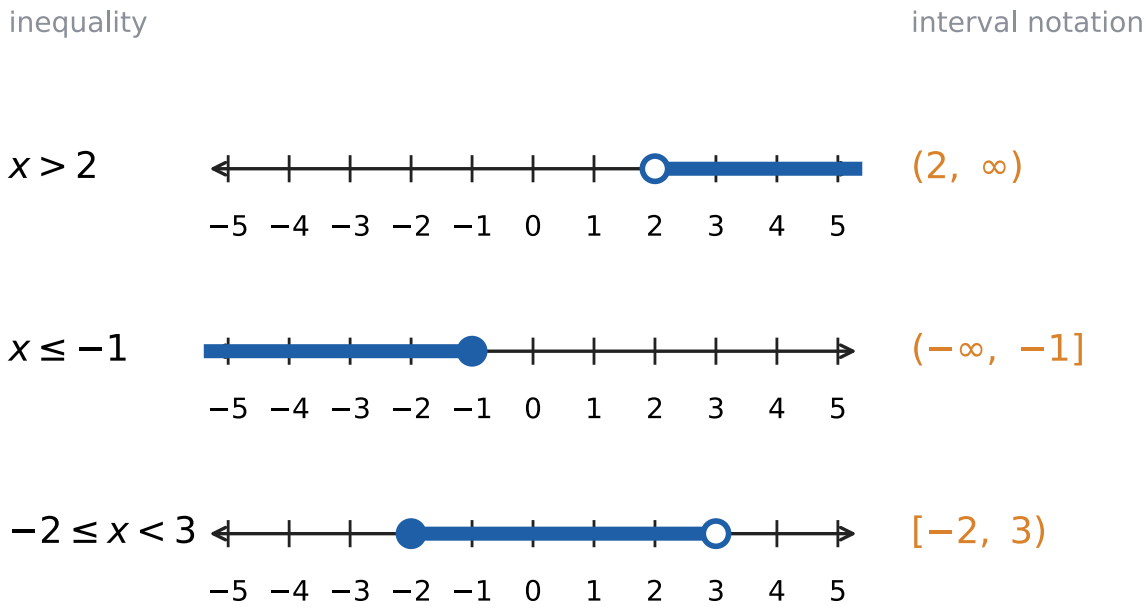


Figure 32. The same three solution sets as inequalities, on the number line, and in interval notation.



Set-builder notation, membership, union and intersection

Set-builder notation $\{x \mid x > 2\}$ is read "the set of all x such that $x > 2$". The vertical bar means "such that" (some books use a colon). $x \in S$ means "x is an element of the set S"; for example $3 \in (2, \infty)$. The **union** $A \cup B$ is everything in A *or* B (or both); the **intersection** $A \cap B$ is everything in *both*. These are developed properly in Volume 3, Chapter 21 .

5.6.8. Compound inequalities

A **compound inequality** joins two inequalities with "and" or "or".

"And": a double inequality. $-3 < 2x + 1 \leq 9$ means $-3 < 2x + 1$ *and* $2x + 1 \leq 9$. Because the middle is sandwiched, you can apply each operation to all three parts at once:

$$\begin{aligned} &-3 < 2x + 1 \leq 9 \\ &-4 < 2x \leq 8 && \text{(subtract 1 from all three parts)} \\ &-2 < x \leq 4 && \text{(divide all three parts by 2)} \end{aligned}$$

Solution set $(-2, 4]$. Check the endpoints: $x = -2$ gives $-3 < -3$, false, so excluded; $x = 4$ gives $9 \leq 9$, true, so included.

When the multiplier is negative, both symbols flip, and it is tidiest to rewrite the result so the smaller number is on the left:

$$\begin{aligned} &-5 \leq 3 - 2x < 7 \\ &-8 \leq -2x < 4 && \text{(subtract 3 throughout)} \\ &4 \geq x > -2 && \text{(divide by } -2\text{: both symbols reverse)} \\ &-2 < x \leq 4 && \text{(rewrite in increasing order)} \end{aligned}$$

"Or": a union. Solve $3x - 1 < -7$ or $2x + 5 \geq 11$. Solve each separately: $3x < -6$ gives $x < -2$; $2x \geq 6$ gives $x \geq 3$. A number satisfies the compound statement if it satisfies *either* part, so the solution set is the union

$$(-\infty, -2) \cup [3, \infty).$$

"And" corresponds to intersection (the overlap of the two solution sets); "or" corresponds to union (everything covered by either). An "and" can be empty: $x < 3$ and $x > 5$ share no numbers, so its solution set is $\varnothing$.



Impossible sandwiches

Never write an "or" as a sandwich. "$x < -2$ or $x \geq 3$" is *not* $3 \leq x < -2$; that sandwich says x is simultaneously at least 3 and less than -2, which is impossible. A double inequality $a < x < b$ only makes sense when $a < b$, and it always means "and".

5.6.9. Where you'd use this

Inequalities answer "how much can I afford?", "how many will fit?" and "what do I need?".

- **Grades.** Your four test scores average 78 and the final test counts the same as each of the others. What score s lifts your average to at least 80? The five scores total $4 \times 78 + s = 312 + s$, so $\frac{312+s}{5} \geq 80$, giving $312 + s \geq 400$ and $s \geq 88$.
- **Engineering limits.** A lift is rated for 1000 kg. The operator weighs 80 kg and each crate 35 kg. The number of crates n must satisfy $80 + 35n \leq 1000$, so $n \leq 26.29\ldots$. Because n is a whole number, the answer is at most 26. Rounding *down* is forced by the direction of the inequality; rounding to the nearest integer would not be safe here.
- **Computing: frame budgets.** A game running at 60 frames per second has $1000/60 \approx 16.67$ ms per frame. If fixed work takes 4 ms and each sprite costs 0.05 ms, the sprite count must satisfy $4 + 0.05n \leq 16.67$, so $n \leq 253$. Performance budgets for web pages, database queries and network requests are the same one-line inequality.

Computer-science connection: bounds checking and half-open intervals

Every array index check is a compound inequality: `0 ⇐ i < n` means $i \in [0, n)$. Python even lets you write it exactly as mathematicians do, because it supports **chained comparisons**: `0 ⇐ i < n` is evaluated as `0 ⇐ i and i < n`. (In C or Java, `0 ⇐ i < n` compiles but means something entirely different: `(0 ⇐ i) < n`, comparing a boolean to `n`.)

The **half-open** interval $[0, n)$ is no accident. `range(0, n)`, slices `a[lo:hi]`, and C++ iterator ranges are all half-open. Half-open intervals tile without overlap or gaps: $[0, 5) \cup [5, 10) = [0, 10)$, and the length of $[a, b)$ is simply $b - a$. Many off-by-one errors are a closed interval treated as half-open, or vice versa.

Solving an inequality for an index is routine in code. Which integers i satisfy $0 \le 3i + 2 < 20$? Subtract 2: $-2 \le 3i < 18$. Divide by 3: $-\frac{2}{3} \le i < 6$. The integers in that range are $0, 1, 2, 3, 4, 5$, i.e. `range(0, 6)`.



Why numbering should start at zero

In a famous one-page note, EWD831 (1982), Edsger Dijkstra argued that the best way to describe the sequence $2, 3, \ldots, 12$ is $2 \le i < 13$, a half-open interval, because adjacent ranges then share an endpoint, the length is the difference of the bounds, and an empty range $a \le i < a$ needs no special case. From that he concluded that indexing should start at zero, so that a sequence of length N is indexed by $0 \le i < N$. The symbols $<$ and $>$ themselves are much older: they first appeared in print in Thomas Harriot's *Artis Analyticae Praxis* (1631).

Computer-science connection: constraints as intervals

Many configuration problems are a handful of linear inequalities in one variable. Each one, solved, is an interval; the values satisfying *all* of them form the intersection of those intervals, the **feasible set**. This is the one-variable version of **constraint satisfaction**, and the one-variable shadow of linear programming (Chapter 6).

PYTHON

```python
import math
from fractions import Fraction

def solve_le(a, b, c):
    """Solution set of a*x + b <= c as a closed interval (lo, hi).

    Returns None if there is no solution.
    """
    a, b, c = Fraction(a), Fraction(b), Fraction(c)
    if a == 0:                          # 0*x + b <= c: always or never true
        return (-math.inf, math.inf) if b <= c else None
    bound = (c - b) / a
    if a > 0:
        return (-math.inf, bound)       # x <= bound
    return (bound, math.inf)            # dividing by a < 0 flips the sign

def intersect(i, j):
    """Intersection of two closed intervals (None means empty)."""
    if i is None or j is None:
        return None
    lo, hi = max(i[0], j[0]), min(i[1], j[1])
    return (lo, hi) if lo <= hi else None

def show(i):
    return "empty" if i is None else f"[{i[0]}, {i[1]}]"

# Constraints on a batch size x, each written as a*x + b <= c:
constraints = [
    (4, 100, 900),   # memory:     4x + 100 <= 900
    (-3, 0, -60),    # throughput: -3x <= -60   (i.e. x >= 20)
    (1, -150, 0),    # API limit:  x - 150 <= 0
]
feasible = (-math.inf, math.inf)
for a, b, c in constraints:
    feasible = intersect(feasible, solve_le(a, b, c))
    print(f"after {a}x + {b} <= {c}:  {show(feasible)}")
```

```
after 4x + 100 <= 900:  [-inf, 200]
after -3x + 0 <= -60:  [20, 200]
after 1x + -150 <= 0:  [20, 150]
```

Notice the branch on the sign of `a`: dividing by a negative `a` flips the inequality, and the sign-flip rule is not a classroom convention but a branch your code must take. Any batch size from 20 to 150 satisfies all three constraints. Add a constraint that contradicts the others and the intersection becomes empty: the system is **infeasible**.



Inequality explorer

A slider sets k; the display shows two points $a < b$ on a number line and their images ka and kb. As k passes through 0 the images collide and then swap order, making the sign-flip rule visible. A second panel lets the learner type an inequality and drag a test point, which turns green inside the solution set and red outside.

5.7. Exercises: inequalities

1. Write in interval notation: (a) $x \geq -3$ (b) $-1 < x \leq 6$ (c) $x < 0$ or $x > 2$.
2. Solve $5 - 2x > 11$ and give the interval.
3. Solve $-4 \leq \dfrac{3x - 1}{2} < 7$.
4. Solve $2x + 1 < -5$ or $3 - x \leq 1$.
5. Explain why "$x < 3$ and $x > 5$" has no solutions, while "$x < 3$ or $x > 5$" has infinitely many.
6. Find every integer i with $-7 < 5 - 4i \leq 13$.

▼ **Answers**

1. (a) $[-3, \infty)$ (b) $(-1, 6]$ (c) $(-\infty, 0) \cup (2, \infty)$.
2. Subtract 5: $-2x > 6$. Divide by -2 and flip: $x < -3$. Interval $(-\infty, -3)$. Test $x = -4$: $5 + 8 = 13 > 11$, true.
3. Multiply all parts by 2: $-8 \leq 3x - 1 < 14$. Add 1: $-7 \leq 3x < 15$. Divide by 3: $-\frac{7}{3} \leq x < 5$. Interval $[-\frac{7}{3}, 5)$.
4. First: $2x < -6$, $x < -3$. Second: $-x \leq -2$, $x \geq 2$. Union: $(-\infty, -3) \cup [2, \infty)$.
5. No number is both below 3 and above 5, so the intersection is empty. The union contains every number below 3 and every number above 5.
6. Subtract 5: $-12 < -4i \leq 8$. Divide by -4, flipping both: $3 > i \geq -2$, i.e. $-2 \leq i < 3$. Integers: $-2, -1, 0, 1, 2$. Check $i = 3$: $5 - 12 = -7$, and $-7 < -7$ is false, so 3 is correctly excluded.

5.8. Absolute-value equations and inequalities

5.8.1. The problem

A machined bolt should be 25.00 mm long, with a tolerance of 0.05 mm. Which lengths L pass inspection? In words: "L is within 0.05 of 25". In symbols:

$$|L - 25| \le 0.05.$$

Absolute values are the language of **distance**, **error** and **tolerance**. They look awkward because the absolute-value function is not linear (it has a corner), but every absolute-value equation or inequality with a linear inside splits into linear pieces you already know how to solve.



Absolute value as distance

The **absolute value** $|x|$ is the distance from x to 0 on the number line, so it is never negative: $|5| = 5$, $|-5| = 5$, $|0| = 0$. Formally,

$$|x| = \begin{cases} x & \text{if } x \ge 0, \\ -x & \text{if } x < 0. \end{cases}$$

(The second line is not a typo: if x is negative, $-x$ is positive.) More usefully, $|x - a|$ is the **distance between x and a**. Read $|x - 3| < 2$ as "x is less than 2 away from 3".

5.8.2. Intuition

Read every absolute-value statement as a distance statement and the answer is almost visible:

- $|x - 3| = 2$: the points at distance exactly 2 from 3, namely 1 and 5.
- $|x - 3| < 2$: the points closer than 2 to 3, namely the open interval $(1, 5)$.
- $|x - 3| > 2$: the points farther than 2 from 3, namely $(-\infty, 1) \cup (5, \infty)$.

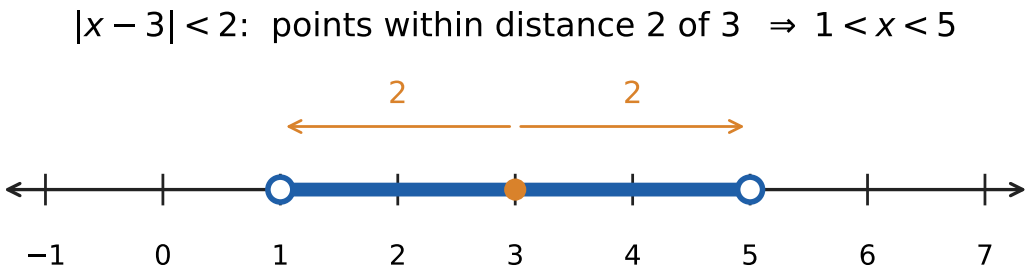


Figure 33. $|x - 3| < 2$ is a distance statement: everything within 2 units of 3.

"Less than" gives one interval in the middle (an "and"); "greater than" gives two outer rays (an "or"). This is the whole theory, and it follows directly from the definition: $|u| = k$ with $k > 0$ means u is either k or $-k$; $|u| < k$ means u is between $-k$ and k.

$$\begin{aligned} |u| = k &\iff u = k \text{ or } u = -k \qquad (k > 0) \\ |u| < k &\iff -k < u < k \\ |u| > k &\iff u < -k \text{ or } u > k \end{aligned}$$

The same holds with $\le$ and $\ge$. If $k < 0$, $|u| = k$ and $|u| < k$ have no solutions (a distance cannot be negative), while $|u| > k$ is true for every u.

5.8.3. The method

Example 31. Procedure — absolute-value equation or inequality with a linear inside

Input: an equation or inequality containing one absolute value $|ax+b|$.

Steps:

1. **Isolate** the absolute value: rearrange to $|ax+b| \;\square\; k$, where $\square$ is $=$, $<$, $\le$, $>$ or $\ge$.
2. **Check the sign of k.** If $k<0$: $=$, $<$, $\le$ have no solution; $>$, $\ge$ hold for all x. If $k=0$, the equation becomes $ax+b=0$.
3. **Split** into linear pieces:
   a. $=$: solve $ax+b=k$ and $ax+b=-k$.
   b. $<$ or $\le$: solve the double inequality $-k < ax+b < k$.
   c. $>$ or $\ge$: solve $ax+b>k$ *or* $ax+b<-k$.
4. Solve each linear piece with the earlier procedures, flipping for negative multipliers.
5. Combine and check.

Output: a set of at most two numbers (equations), an interval ("less than"), or a union of two rays ("greater than").

5.8.4. Worked examples

Example 1. Solve $3|x-1|+2=14$.

$$\begin{aligned} 3|x-1| &= 12 && \text{(subtract 2)}\\ |x-1| &= 4 && \text{(divide by 3; absolute value now isolated)}\\ x-1=4 \quad &\text{or} \quad x-1=-4 && \text{(split)}\\ x=5 \quad &\text{or} \quad x=-3 \end{aligned}$$

Check: $3|4|+2=14$ and $3|-4|+2=14$. Both work. (Geometrically: the two points at distance 4 from 1.)

Example 2. Solve $|2x-5|=7$.

$2x-5=7$ gives $x=6$; $2x-5=-7$ gives $2x=-2$, $x=-1$. Check: $|7|=7$ and $|-7|=7$.

Example 3. Solve $|2x+1|\ge 5$.

$$\begin{aligned} 2x+1 \ge 5 \quad &\text{or} \quad 2x+1 \le -5 && \text{(greater-than: two outer pieces)}\\ 2x \ge 4 \quad &\text{or} \quad 2x \le -6\\ x \ge 2 \quad &\text{or} \quad x \le -3 \end{aligned}$$

Solution set $(-\infty,-3]\cup[2,\infty)$. Test $x=0$, which should fail: $|1|=1\ge 5$ is false. Good.

Example 4. The bolt: $|L - 25| \le 0.05$ becomes $-0.05 \le L - 25 \le 0.05$, so $24.95 \le L \le 25.05$.

5.8.5. Alternative approaches

- **Distance reading.** When the inside is $x - a$ (coefficient 1), skip the algebra: $|x + 4| < 3$ is $|x - (-4)| < 3$, "within 3 of -4", so $(-7, -1)$.
- **Case analysis on the sign of the inside.** Use the piecewise definition: if $2x - 5 \ge 0$ then $|2x - 5| = 2x - 5$; otherwise it is $-(2x - 5)$. Solve in each case and keep only solutions consistent with the case. Slower, but it is the only method that works for equations with *two* absolute values or with x outside too, such as $|x - 1| = 2x + 4$.
- **Squaring.** Since both sides are non-negative, $|u| = k \Leftrightarrow u^2 = k^2$. This produces a quadratic, so it is heavier machinery; but it is the natural method for $|x - 1| = |x - 5|$, where squaring both sides cancels the x^2 terms: $x^2 - 2x + 1 = x^2 - 10x + 25$, so $8x = 24$, $x = 3$. (Geometrically: the point equidistant from 1 and 5 is the midpoint 3.)

Choosing: use the distance reading for $|x - a|$ forms, the split procedure for a single $|ax + b|$ isolated on one side, and case analysis whenever x also appears outside the absolute value. Whenever x appears outside, check every candidate in the original equation, because the right side might be negative.



Absolute-value traps

- **Isolate first.** In $3|x - 1| + 2 = 14$ you cannot split into $3(x - 1) + 2 = \pm 14$; that gives wrong answers. Get $|x - 1|$ alone first.
- **Negative right side.** $2|x - 4| + 3 = 1$ gives $|x - 4| = -1$: no solution. Splitting blindly would "find" $x = 3$ and $x = 5$, both of which fail the check.
- **"Less-than" is "and"; "greater-than" is "or".** Writing $|2x + 1| \ge 5$ as the sandwich $-5 \le 2x + 1 \le 5$ solves the opposite problem.
- $|a + b| \ne |a| + |b|$ in general: $|3 + (-5)| = 2$ but $|3| + |-5| = 8$. You cannot "distribute" an absolute value over a sum.

5.8.6. Where you'd use this

Any range $[a, b]$ can be written as one absolute-value inequality: centre $c = \frac{a+b}{2}$, half-width $r = \frac{b-a}{2}$, and $a \le x \le b \iff |x - c| \le r$. That is how specifications, tolerances and error bars are quoted.

- **Health.** A target range of 4 to 7 mmol/L for blood glucose is $|G - 5.5| \le 1.5$: "within 1.5 of 5.5".
- **Science.** A measurement reported as $g = 9.81 \pm 0.02\ \mathrm{m/s^2}$ means $|g - 9.81| \le 0.02$. Two experiments agree if their ranges overlap; with 9.81 ± 0.02 and 9.86 ± 0.02 they do not, since $|9.86 - 9.81| = 0.05 > 0.02 + 0.02$.
- **Manufacturing.** The bolt above: $|L - 25| \le 0.05$ is the interval $[24.95, 25.05]$ mm.

- **Games and graphics.** Two boxes overlap horizontally exactly when the distance between their centres is less than the sum of their half-widths: $|c_1 - c_2| < \frac{w_1 + w_2}{2}$. With centres at 10 and 16 and widths 4 and 6, $|10 - 16| = 6$ is not less than 5, so there is no collision. Do the same test in y and you have the axis-aligned bounding-box (AABB) check at the heart of most collision detection.

Computer-science connection: comparing floating-point numbers

Because floating-point arithmetic rounds, `0.1 + 0.2 == 0.3` is `False` in Python. The correct test is an absolute-value inequality: $|a - b| \le \varepsilon$ for some tolerance ε (the Greek letter epsilon, conventionally a small positive number). Python's `math.isclose(a, b, rel_tol=1e-09, abs_tol=0.0)` uses a **relative** version, $|a - b| \le \varepsilon \cdot \max(|a|, |b|)$, so the tolerance scales with the size of the numbers. Test suites, physics engines and convergence checks in numerical algorithms (Volume 2, Chapter 32) all hinge on such inequalities. Absolute error $|x - x_{\text{true}}|$ is also the building block of the **L1 loss** and **mean absolute error** in machine learning.

5.9. Exercises: absolute value

1. Solve $|x + 2| = 5$.
2. Solve $|3x - 6| \le 9$.
3. Solve $|1 - 2x| > 3$.
4. Solve $2|x - 4| + 3 = 1$.
5. Write "x is within 0.5 of 7" as an absolute-value inequality and as an interval.
6. Solve $|x - 1| = |x - 5|$ by splitting rather than squaring. Interpret the answer as a distance.

▼ **Answers**

1. $x + 2 = 5$ or $x + 2 = -5$: $x = 3$ or $x = -7$. (The points at distance 5 from -2.)
2. $-9 \le 3x - 6 \le 9$; add 6: $-3 \le 3x \le 15$; divide by 3: $-1 \le x \le 5$. Interval $[-1, 5]$.
3. $1 - 2x > 3$ or $1 - 2x < -3$. First: $-2x > 2$, $x < -1$. Second: $-2x < -4$, $x > 2$. Solution $(-\infty, -1) \cup (2, \infty)$.
4. $2|x - 4| = -2$, so $|x - 4| = -1$. An absolute value cannot be negative: no solution, $\varnothing$.
5. $|x - 7| \le 0.5$, i.e. $[6.5, 7.5]$.
6. Two absolute values are equal when the insides are equal or opposite. $x - 1 = x - 5$ gives $-1 = -5$, impossible. $x - 1 = -(x - 5)$ gives $2x = 6$, $x = 3$. The point equidistant from 1 and 5 is their midpoint, 3.

5.10. The straight line: $y = mx + b$

5.10.1. The problem

So far the unknown has been a single number to be found. Step back and ask a bigger question about the taxi: how does the fare depend on the distance, *for every* distance? Tabulate it:

distance d (miles)	**0**	**1**	**2**	**3**	**4**
fare C (\$)	3.00	5.50	8.00	10.50	13.00

Each extra mile adds the same \$2.50. That constant step is the defining feature of a **linear relationship**, and it is why the graph is a straight line. Every equation you solved earlier in this chapter was a question about a line; seeing the line makes the algebra visible.

5.10.2. Intuition: constant rate of change

A relationship $y = mx + b$ has two numbers that say everything about it.

- b is the **starting value**: the value of y when $x = 0$. For the taxi, $b = 3$, the flag-fall fee.
- m is the **rate of change**: how much y changes for each unit increase in x. For the taxi, $m = 2.5$ dollars per mile.

Rate of change is the idea to hold on to. Speed is a rate of change (miles per hour), price is a rate of change (dollars per kilogram), throughput is a rate of change (requests per second). A linear model says the rate is the same everywhere. Calculus (Volume 2, Chapter 17) is, in large part, about what to do when it is not.



Slope, gradient, intercept

In $y = mx + b$ (the **slope–intercept form**):

- m is the **slope** (in UK usage, the **gradient**): the change in y per unit change in x. Positive slope rises to the right; negative slope falls; zero slope is horizontal.
- b is the **y-intercept**: the line crosses the y-axis at $(0, b)$. (UK texts often write $y = mx + c$.)
- The **x-intercept** is where the line crosses the x-axis, found by solving $mx + b = 0$, so $x = -b/m$ when $m \neq 0$.

The slope has **units**: units of y per unit of x.



Delta notation

Δ (capital Greek delta) means "change in". $\Delta x = x_2 - x_1$ is the change in x from a first point (x_1, y_1) to a second (x_2, y_2); the subscripts 1 and 2 simply label the points. $\Delta y = y_2 - y_1$. Slope is "rise over run":

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}, \qquad x_1 \neq x_2.$$

5.10.3. Why it works: the slope is the same everywhere

Why does $y = mx + b$ give a straight line, and why is the slope the same whichever two points you pick? Take any two points on the graph, at x_1 and x_2. Then

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{(mx_2 + b) - (mx_1 + b)}{x_2 - x_1} = \frac{m(x_2 - x_1)}{x_2 - x_1} = m.$$

The b cancels and the ratio is always m. So every "rise over run" triangle drawn on the graph has the same shape (the triangles are **similar**, in the language of Chapter 9), and a curve whose rise-over-run triangles all have the same shape cannot bend: it is a straight line. Conversely, any non-vertical line has a constant ratio $\Delta y / \Delta x$, which is exactly what $y = mx + b$ encodes.

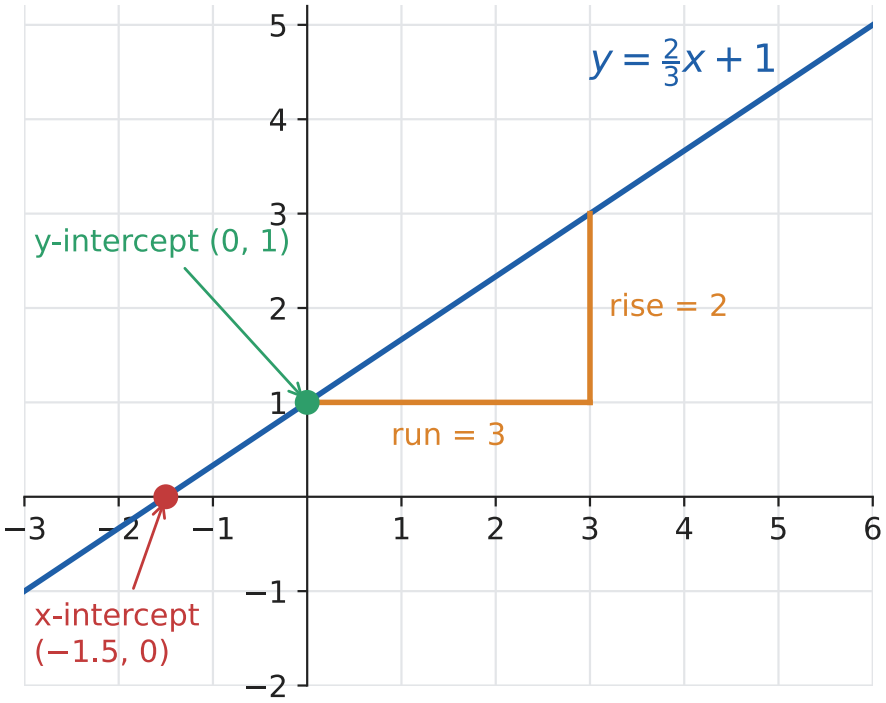


Figure 34. The line $y = \frac{2}{3}x + 1$: a run of 3 produces a rise of 2, so the slope is $\frac{2}{3}$. The y-intercept is $(0, 1)$; the x-intercept solves $\frac{2}{3}x + 1 = 0$.

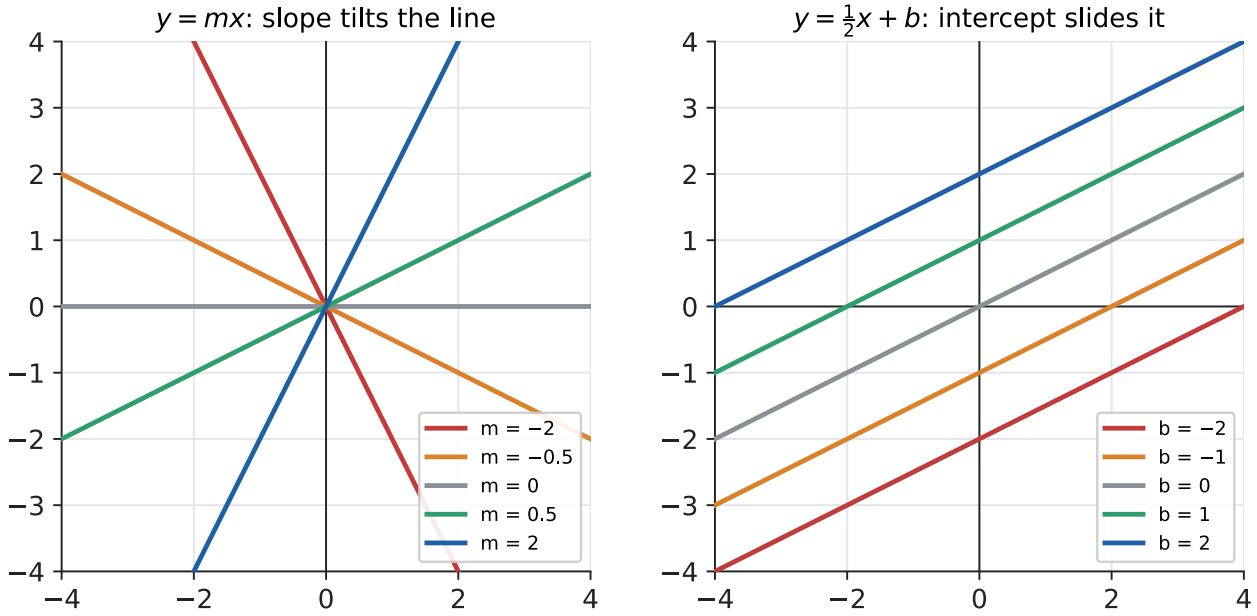


Figure 35. Changing m rotates the line about its intercept; changing b slides it up and down without changing its direction.

5.10.4. Three forms of a line's equation

The same line can be written in several forms, each convenient for a different job.

Form	Equation	Best when you know…
slope–intercept	$y = mx + b$	the slope and the y-intercept; for graphing and evaluating
point–slope	$y - y_1 = m(x - x_1)$	the slope and any one point (x_1, y_1)
standard (general)	$Ax + By = C$	nothing in particular; it treats x and y symmetrically and is the form used for systems (Chapter 6)

Point–slope form is just the slope definition rearranged: if (x, y) is any other point on the line, then $\frac{y - y_1}{x - x_1} = m$; multiply both sides by $(x - x_1)$.

Example 32. Procedure — equation of the line through two points

Input: two points (x_1, y_1) and (x_2, y_2).

Steps:

1. If $x_1 = x_2$, the line is vertical: its equation is $x = x_1$. Stop.
2. Compute the slope $m = \dfrac{y_2 - y_1}{x_2 - x_1}$, subtracting in the *same order* top and bottom.
3. Substitute into point–slope form: $y - y_1 = m(x - x_1)$.
4. Expand and add y_1 to get slope–intercept form $y = mx + b$.
5. Check that the *other* point satisfies the equation.

Output: the equation of the line.

5.10.5. Worked example

Find the line through $(2, 1)$ and $(6, 7)$, and its intercepts.

$$\begin{aligned} m &= \frac{7 - 1}{6 - 2} = \frac{6}{4} = \frac{3}{2} && \text{(slope)} \\ y - 1 &= \tfrac{3}{2}(x - 2) && \text{(point–slope, using } (2, 1)\text{)} \\ y - 1 &= \tfrac{3}{2}x - 3 && \text{(expand)} \\ y &= \tfrac{3}{2}x - 2 && \text{(add 1)} \end{aligned}$$

Check with the other point: $\frac{3}{2} \cdot 6 - 2 = 9 - 2 = 7$. Correct.

The y-intercept is $(0, -2)$. The x-intercept solves a linear equation:

$$\tfrac{3}{2}x - 2 = 0 \implies \tfrac{3}{2}x = 2 \implies x = \tfrac{4}{3}.$$

In standard form, multiply $y = \frac{3}{2}x - 2$ by 2 and rearrange: $3x - 2y = 4$.

5.10.6. Special lines, parallel and perpendicular

- **Horizontal** lines have slope 0: $y = k$. The output never changes.
- **Vertical** lines have undefined slope (the run is zero, and division by zero is undefined): $x = k$. A vertical line is not of the form $y = mx + b$ at all, which is one reason standard form $Ax + By = C$ exists: it covers vertical lines with $B = 0$.
- **Parallel** lines have equal slopes. The line through $(2, 3)$ parallel to $y = 4x - 1$ is $y - 3 = 4(x - 2)$, i.e. $y = 4x - 5$.
- **Perpendicular** lines (neither vertical) have slopes whose product is -1: $m_1 m_2 = -1$. So a line perpendicular to slope $\frac{3}{2}$ has slope $-\frac{2}{3}$. Why this is so is a pleasant piece of geometry, done in Chapter 10.

5.10.7. Equations and inequalities, seen on the line

Here is the payoff: every technique in this chapter has a picture.

- Solving $mx + b = 0$ is finding where the line crosses the x-axis.
- Solving $mx + b = k$ is finding where the line meets the horizontal line $y = k$.
- Solving $mx + b > 0$ is finding where the line is *above* the axis. A rising line ($m > 0$) is above the axis to the right of its x-intercept; a falling line ($m < 0$) is above it to the *left*. That is the sign-flip rule again, now as a picture: dividing by a negative slope reverses the direction of the answer.
- Solving $ax + b = cx + d$ is finding where two lines cross; the "no solution" and "all solutions" cases are parallel and coincident lines.
- Solving $|x - a| < r$ is finding where the V-shaped graph $y = |x - a|$ dips below the horizontal line $y = r$.

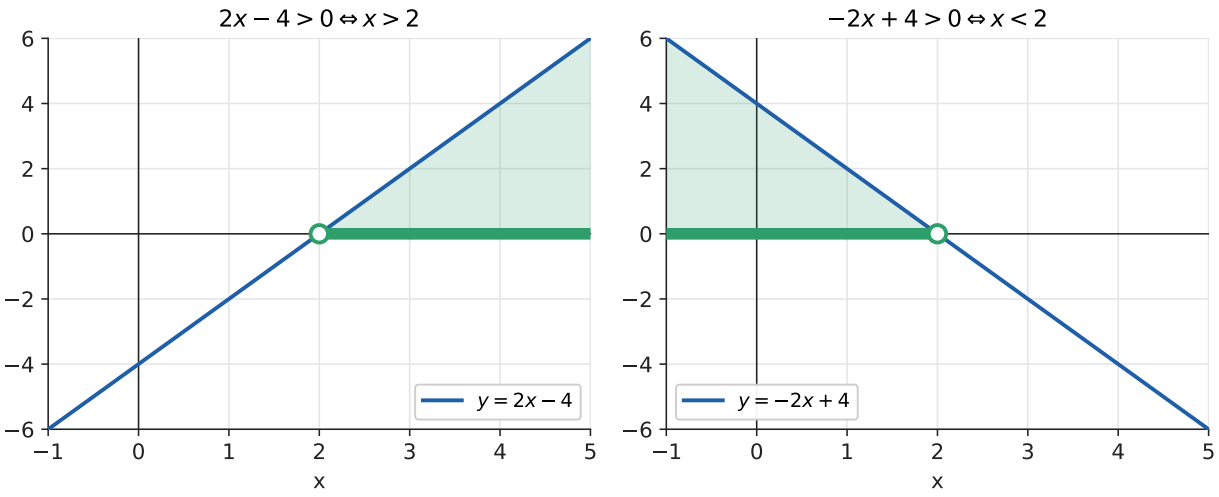


Figure 36. Solving $mx + b > 0$ by eye: the solution set (green on the x-axis) is where the line is above the axis. A rising line is above it to the right of the intercept; a falling line, to the left. The open circle marks the excluded boundary $x = 2$.



Slope pitfalls

- **Inconsistent order.** $\frac{7-1}{2-6}$ (top from point 2 minus point 1, bottom the other way round) gives $-\frac{3}{2}$, the wrong sign. Subtract in the same order top and bottom.
- **Run over rise.** Slope is change in y over change in x, never the reverse.
- **Reading slopes off a picture.** A graph whose axes have different scales distorts the apparent steepness; compute from coordinates.
- **Linear is a model, not a law.** Taxi fares are linear by design; most real relationships are only approximately linear over a limited range. Extrapolating far outside the observed data is risky (see Section 5.13).

5.10.8. Where you'd use this

A slope is a **rate**: "so much output per unit of input". Once you look for it, it is everywhere.

- **Travel and building.** A road sign reading "10%" means a slope of 0.1: 10 m of climb per 100 m horizontally. Accessibility guidelines in the US and UK limit wheelchair ramps to about 1 in 12 (slope $\frac{1}{12}$), so a 0.6 m step needs a ramp at least $0.6 \times 12 = 7.2$ m long.
- **Mechanics.** On a velocity–time graph the slope is the acceleration and the intercept is the initial velocity. A car doing 4 m/s at $t = 0$ and 16 m/s at $t = 6$ s has slope $\frac{16-4}{6-0} = 2\ \mathrm{m/s^2}$, so $v = 4 + 2t$: exactly $v = u + at$. Hooke's law ($F = kx$) and Ohm's law ($V = IR$) are lines through the origin whose slopes are the spring constant and the resistance.
- **Computing.** The simplest machine-learning model, linear regression with one input, is $y = mx + b$: the training data choose the slope (the weight) and the intercept (the bias). A single artificial neuron is the same formula before its activation function (Volume 3, Chapter 33).

Computer-science connection: linear time

When an algorithm is "linear time", $O(n)$, its running time is modelled as $T(n) \approx mn + b$: a fixed overhead b plus a cost m per item. Benchmark at two sizes and you can estimate m and b exactly as in the two-point procedure. With many noisy measurements, the "best" line through them is found by **linear regression** (Volume 3, Chapter 31), which is ultimately a system of linear equations (Chapter 6).



Who chose m?

Nobody knows why American textbooks use m for slope. Popular explanations (French *monter*, "to climb"; "modulus of slope") are unsupported by evidence; the letter appears in 19th-century English and American texts without explanation. British texts use $y = mx + c$, some European ones $y = kx + d$ or $y = ax + b$. The idea matters; the letter does not.



Slope and intercept sliders

Sliders for m and b redraw $y = mx + b$ live, with a draggable rise–run triangle whose size changes while its ratio stays fixed. A second mode shows two draggable points; the line through them and its three equation forms update as the points move. A third mode shades the region where $mx + b > k$ on the x-axis.

Try it: Slide m and b

5.11. Exercises: lines

1. Find the slope, y-intercept and x-intercept of $3x + 2y = 12$.
2. Find the equation of the line through $(-1, 4)$ and $(3, -4)$.
3. Find the line through $(2, 3)$ perpendicular to $y = \frac{1}{2}x + 7$.
4. A taxi charges \$3.50 plus \$2.20 per mile. Write the fare as a linear function of distance, and find the greatest distance you can afford with \$25.
5. A table shows $x = 0, 2, 4$ and $y = 5, 11, 17$. Is the relationship linear? If so, find its equation and decide whether $(10, 34)$ fits.

▼ **Answers**

1. $2y = -3x + 12$, so $y = -\frac{3}{2}x + 6$: slope $-\frac{3}{2}$, y-intercept $(0, 6)$. Setting $y = 0$: $3x = 12$, x-intercept $(4, 0)$.
2. $m = \frac{-4-4}{3-(-1)} = \frac{-8}{4} = -2$. $y - 4 = -2(x + 1)$, so $y = -2x + 2$. Check $(3, -4)$: $-6 + 2 = -4$.
3. The perpendicular slope is $-1 \div \frac{1}{2} = -2$. $y - 3 = -2(x - 2)$, so $y = -2x + 7$.

4. $C = 3.5 + 2.2d$. Solve $3.5 + 2.2d \leq 25$: $2.2d \leq 21.5$, $d \leq 21.5/2.2 \approx 9.77$. So up to about 9.77 miles (with $d \geq 0$).
5. Each step of 2 in x adds 6 to y: constant rate, so linear, with $m = 3$ and $b = 5$: $y = 3x + 5$. At $x = 10$, $y = 35$, so $(10, 34)$ does not fit.

5.12. Modelling: word problems as linear equations

5.12.1. The problem

Real problems do not arrive as equations. They arrive as sentences, tables, invoices, and requirements. The mathematical skill is **translation**: deciding what the unknown is, expressing every quantity in terms of it, and recognising which statement is the equation. The solving, by now, is the easy part.

Example 33. Procedure — linear modelling

Input: a problem described in words.

Steps:

1. **Name the unknown** in a full sentence with units: "Let t be the number of hours after 9:00 that the first cyclist has been riding." Vague definitions ("let t = time") cause most errors.
2. **Express every other quantity** in terms of the unknown, keeping units consistent (do not mix minutes with hours, or percentages with fractions).
3. **Find the equality or inequality** the problem asserts ("catches up" means equal distances; "is cheaper" means one cost is less than the other).
4. **Solve** with the procedures of this chapter.
5. **Check** the answer against the original words, not just the equation.
6. **Interpret**: answer the question actually asked, with units and a sentence.

Output: the answer, in context.

5.12.2. Worked example 1: rates (catching up)

Cyclist A leaves at 9:00 riding at 12 km/h. Cyclist B leaves from the same place at 9:30 on the same route at 18 km/h. When and where does B catch A?

Let t be the number of hours after 9:00. A has ridden for t hours and B for $t - 0.5$ hours (30 minutes later, in hours). Distance is speed times time, and "catches" means equal distances:

$$\begin{aligned} 12t &= 18(t - 0.5) \\ 12t &= 18t - 9 && \text{(expand)} \\ 9 &= 6t && \text{(add 9, subtract } 12t\text{)} \\ t &= 1.5 && \text{(divide by 6)} \end{aligned}$$

Check: by then A has ridden 1.5 h at 12 km/h, i.e. 18 km; B has ridden 1 h at 18 km/h, i.e. 18 km. **Interpret:** B catches A at 10:30, 18 km from the start.

5.12.3. Worked example 2: mixtures

How many litres of a 30% acid solution must be added to 10 litres of a 10% solution to obtain a 25% solution?

Let x be the litres of 30% solution added. The key is to track the quantity that is conserved: **pure acid**.

	litres	acid fraction	litres of acid
10% solution	10	0.10	1
30% solution	x	0.30	$0.3x$
mixture	$10 + x$	0.25	$0.25(10 + x)$

Acid in = acid out:

$$\begin{aligned} 1 + 0.3x &= 0.25(10 + x) \\ 1 + 0.3x &= 2.5 + 0.25x && \text{(expand)} \\ 100 + 30x &= 250 + 25x && \text{(multiply by 100 to clear decimals)} \\ 5x &= 150 && \text{(collect)} \\ x &= 30 \end{aligned}$$

Check: 40 litres containing $1 + 9 = 10$ litres of acid is $10/40 = 25\%$. **Interpret:** add 30 litres. The same table structure solves blends of coffee by price, alloys by purity, and investments by interest rate: a weighted average is a linear equation.

5.12.4. Worked example 3: cost models and break-even

A mobile provider offers Plan A at \$20 per month plus \$3 per GB, and Plan B at \$50 per month plus \$1 per GB. When is each plan cheaper?

Let g be the GB used in a month. The costs are $C_A = 20 + 3g$ and $C_B = 50 + g$. The **break-even point** is where they are equal:

$$\begin{aligned} 20 + 3g &= 50 + g \\ 2g &= 30 && \text{(subtract } g \text{ and } 20\text{)} \\ g &= 15 \end{aligned}$$

At 15 GB both cost $20 + 45 = 65$ dollars. Plan A is cheaper when $20 + 3g < 50 + g$, i.e. $2g < 30$, i.e. $g < 15$. **Interpret:** below 15 GB a month choose A, above 15 GB choose B.

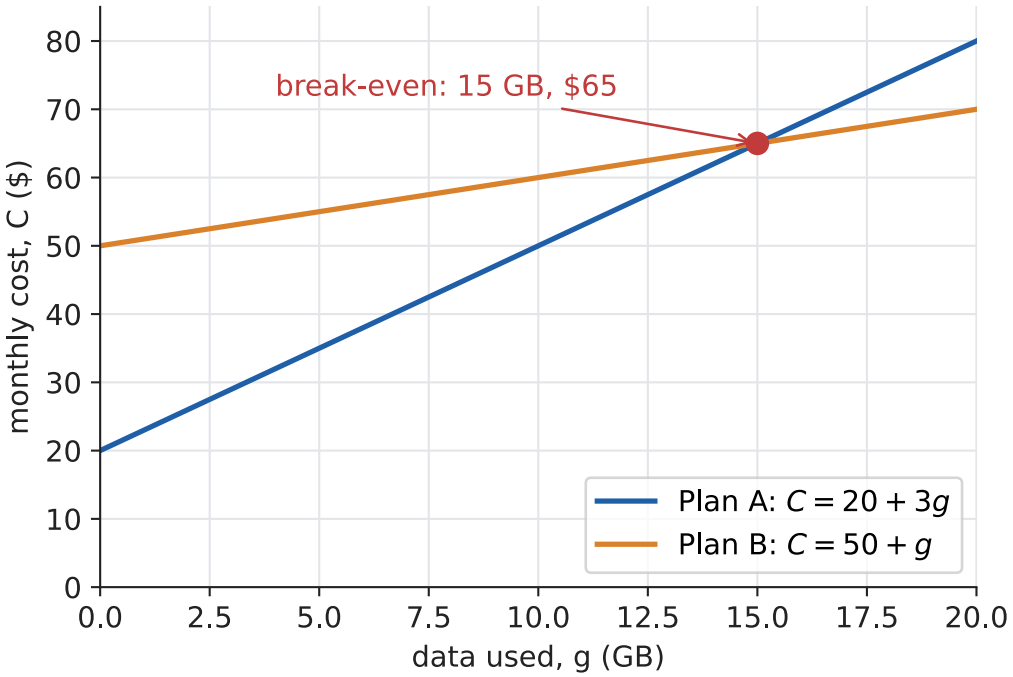


Figure 37. Two linear cost models. Plan A has the lower fixed cost; Plan B the lower marginal cost. They cross at the break-even point.

This shape, a **fixed cost** plus a **marginal cost** per unit, is the single most common model in business, and the comparison is always the same: the plan with higher fixed cost and lower marginal cost wins at high volume. (Strictly, finding the crossing of two lines is a system of two equations; Chapter 6 develops that view.)



Modelling pitfalls

- **Units.** "30 minutes later" is 0.5 hours if speeds are in km/h, not 30.
- **Percentages.** 30% enters the equation as 0.3, and you take 30% *of the right quantity* (of the 30% solution, not of the mixture).
- **Answering the wrong question.** In the cyclist problem, $t = 1.5$ is not the answer; "10:30, 18 km out" is.
- **Model validity.** A linear model is only as good as its assumptions. Constant speed, constant price per unit and no volume discounts are idealisations. Always ask over what range the model is believable.

Computer-science connection: linear cost models in the cloud

Cloud pricing is linear models all the way down. A reserved server at $200/month versus on-demand at $0.40/hour breaks even where $0.4h = 200$, i.e. at $h = 500$ hours a month; since a month is roughly 730 hours, an always-on service should be reserved and a service running under 500 hours should not. The same reasoning compares serverless (no fixed cost, high per-request cost) with dedicated machines (high fixed cost, low marginal cost), build versus buy, and caching (pay fixed memory to save a per-request computation). Amortised analysis of algorithms has the same flavour: a fixed setup cost spread over many operations.



Where Celsius meets Fahrenheit

Temperature conversion is linear: $F = \frac{9}{5}C + 32$. Is there a temperature at which both scales read the same? Set $F = C$: $C = \frac{9}{5}C + 32$, so $-\frac{4}{5}C = 32$, so $C = -40$. At $-40°$ the two scales agree, which is why Arctic weather reports sometimes do not bother to say which scale they mean. Two lines with different slopes always meet exactly once.

5.13. Linear interpolation

5.13.1. The problem

A weather station logs 12 °C at 10:00 and 20 °C at 14:00. What was the temperature at 11:30? An animation moves a sprite from $x = 100$ at frame 0 to $x = 400$ at frame 60. Where is it at frame 45? A lookup table gives a sensor's calibration at a few points; what does a reading between two entries mean?

All three ask for a value *between* two known values, assuming the quantity changes at a constant rate in between. That is **linear interpolation**, known in graphics and game programming as **lerp**.

5.13.2. Intuition and derivation

Interpolating means walking along the straight line from one known value a to the other, b. Describe your position by the **fraction of the way** you have gone, t: $t = 0$ at the start, $t = 1$ at the end, $t = 0.5$ halfway. The total change is $b - a$, so after a fraction t of the journey you have covered $t(b - a)$ of it:

$$\operatorname{lerp}(a, b, t) = a + t\,(b - a) = (1 - t)\,a + t\,b.$$

The two forms are equal by expanding: $a + tb - ta = (1 - t)a + tb$. The second form shows lerp as a **weighted average**: weight $1 - t$ on a, weight t on b, weights summing to 1. It works unchanged for points, colours, and vectors: lerp each coordinate.

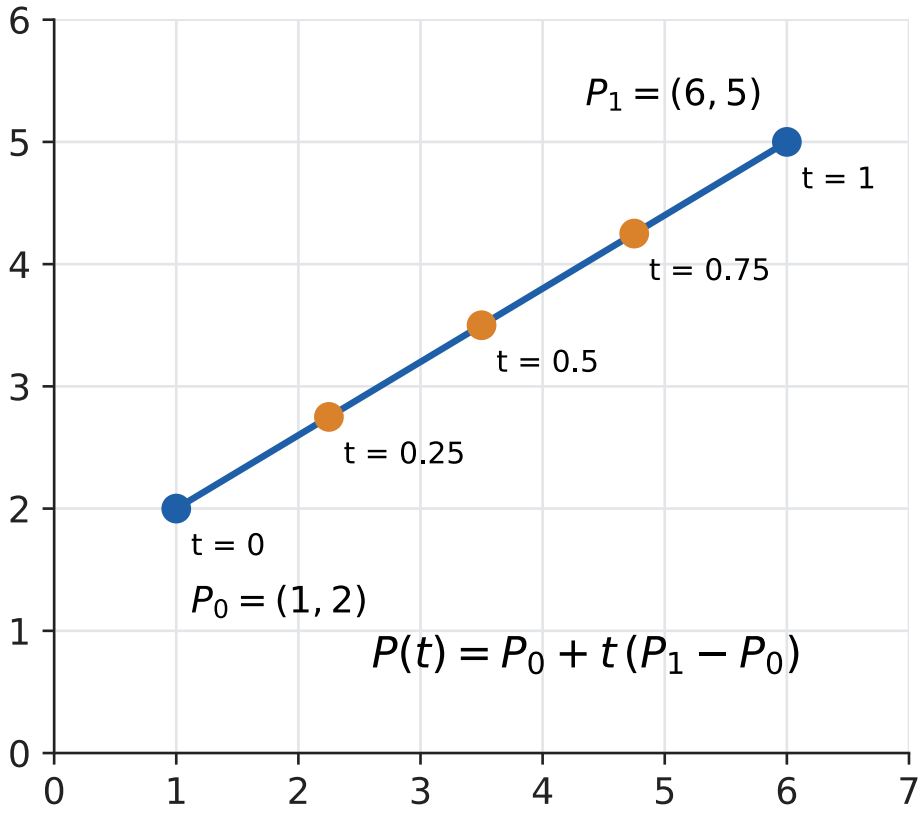


Figure 38. Linear interpolation between two points. The parameter t is the fraction of the way from P_0 to P_1.

The **inverse** question, "what fraction of the way is the value v?", is a linear equation in t:

$$\begin{aligned} a + t(b - a) &= v \\ t(b - a) &= v - a && \text{(subtract } a\text{)} \\ t &= \frac{v - a}{b - a} && \text{(divide by } b - a \neq 0\text{)} \end{aligned}$$

Combine the two and you can **remap** a value from one range $[a, b]$ to another $[c, d]$: find t in the first range, then lerp in the second.

Example 34. Procedure — linear interpolation

Input: two known data points (x_0, y_0) and (x_1, y_1) with $x_0 \neq x_1$, and a query x.

Steps:

1. Compute the fraction $t = \dfrac{x - x_0}{x_1 - x_0}$ (inverse lerp).
2. If $t < 0$ or $t > 1$, you are **extrapolating**, not interpolating; decide whether that is acceptable, or clamp t to $[0, 1]$.
3. Compute $y = y_0 + t\,(y_1 - y_0)$ (lerp).

Output: the estimate y at x.

5.13.3. Worked example

Temperature at 11:30, given 12 °C at 10:00 and 20 °C at 14:00. Measure time in hours: $x_0 = 10$, $x_1 = 14$, $x = 11.5$.

$$t = \frac{11.5 - 10}{14 - 10} = \frac{1.5}{4} = 0.375 \qquad \text{(fraction of the way)}$$
$$y = 12 + 0.375\,(20 - 12) = 12 + 0.375 \cdot 8 = 12 + 3 = 15 \qquad \text{(lerp)}$$

The estimate is 15 °C. This is exactly the same as finding the line through $(10, 12)$ and $(14, 20)$ (slope 2 °C per hour) and evaluating it at 11.5: $12 + 2 \cdot 1.5 = 15$. Interpolation is just "the equation of the line through two points", packaged for repeated use.

```python
def lerp(a, b, t):
    """Point a fraction t of the way from a to b (t=0 -> a, t=1 -> b)."""
    return a + t * (b - a)

def inverse_lerp(a, b, v):
    """Which t gives lerp(a, b, t) == v?  Solves a linear equation for t."""
    if a == b:
        raise ValueError("a and b must differ")
    return (v - a) / (b - a)

def remap(v, in_lo, in_hi, out_lo, out_hi):
    """Map v from one range to another, linearly."""
    return lerp(out_lo, out_hi, inverse_lerp(in_lo, in_hi, v))

print(lerp(10, 30, 0.25))             # 15.0
print(inverse_lerp(10, 30, 15))       # 0.25
print(remap(25, 0, 100, 32, 212))     # 25 C in F -> 77.0
print(remap(512, 0, 1023, 0, 3.3))    # 10-bit ADC reading -> volts
```

```
15.0
0.25
77.0
1.6516129032258065
```

`remap` is the Celsius–Fahrenheit conversion in disguise (0 °C ↦ 32 °F, 100 °C ↦ 212 °F) and the standard way to turn a raw sensor reading into physical units.

5.13.4. Alternatives, choices and failure modes

Linear interpolation is the simplest member of a family. When data are smooth and you have more than two points, **polynomial** or **spline** interpolation (Chapter 7, Volume 2, Chapter 32) follows curves better; when data are noisy, **fitting** a line by regression (Volume 3, Chapter 31) beats passing exactly through two noisy points. Choose lerp when the points are close together relative to how quickly the quantity curves, when speed matters (it is one multiply and two adds), or when "straight-line between keyframes" is the behaviour you actually want.



Interpolation pitfalls

- **Extrapolation.** With t outside $[0, 1]$, the formula still returns a number, but you are trusting the straight line beyond the data. Twelve degrees at 10:00 and twenty at 14:00 does not mean 40 °C at midnight. Clamp t when you mean to stay in range.
- **Floating-point endpoints.** $a + t(b - a)$ can fail to return exactly b at $t = 1$ because of rounding, while $(1 - t)a + tb$ returns b exactly at $t = 1$ but may not be monotonic in t. Library implementations (for example C++20's `std::lerp`) take care over such cases.
- **Nonlinear quantities.** Interpolating angles across $360°$, or colours in a non-linear colour space, or values that grow exponentially, gives visibly wrong results. Lerp the right representation.

Computer-science connection: lerp everywhere

Lerp is among the most-called functions in graphics and games. GPUs perform it in hardware when texture sampling (**bilinear filtering** is lerp in two directions); animation systems lerp between keyframes; colour gradients lerp RGB channels; audio resamplers lerp between samples; "easing" curves just replace t by a nonlinear function of t before lerping. Binary search on sorted numeric data can be upgraded to **interpolation search**, which guesses position $t = \frac{v - a}{b - a}$ within the current range, giving $O(\log \log n)$ average time on uniformly distributed keys.



Curves made of straight lines

The smooth Bézier curves in every font and vector drawing are built from nothing but lerps. Given three control points P_0, P_1, P_2, lerp $P_0 \to P_1$ and $P_1 \to P_2$ at the same t, then lerp between those two results at t again. As t runs from 0 to 1 the final point traces a parabola. This is **de Casteljau's algorithm**, developed by Paul de Casteljau at Citroën around 1959 for designing car bodies.



Lerp playground

Drag two points and a slider for t from -0.5 to 1.5; the interpolated point moves along the line, turning red when it leaves $[0, 1]$ (extrapolation). A second panel adds a third control point and shows the nested lerps of de Casteljau's construction tracing a Bézier curve.

5.14. Exercises: modelling and interpolation

1. A rectangle has perimeter 46 cm and its length is 5 cm more than its width. Find its dimensions.

2. A shop blends coffee costing \$12/kg with coffee costing \$20/kg to make 40 kg of a blend selling at \$15/kg (with no change in total value). How much of each?
3. A gym charges a \$40 joining fee and \$25 a month. A rival charges no joining fee and \$33 a month. After how many whole months is the first gym cheaper in total?
4. A colour channel fades from 40 to 240. What is its value at $t = 0.3$? At what t does it reach 190?
5. A 10-bit temperature sensor reports integers 0 to 1023, mapping linearly onto -40 °C to 125 °C. What temperature does a reading of 620 represent?

▼ **Answers**

1. Let w be the width in cm; length $w + 5$. $2(w + w + 5) = 46$, so $4w + 10 = 46$, $w = 9$. Dimensions 9 cm by 14 cm. Check: $2(9 + 14) = 46$.
2. Let x be the kg at \$12; then $40 - x$ kg at \$20. Value: $12x + 20(40 - x) = 15 \cdot 40 = 600$. Expand: $12x + 800 - 20x = 600$, so $-8x = -200$, $x = 25$. Blend 25 kg of the \$12 coffee with 15 kg of the \$20 coffee. Check: $300 + 300 = 600$.
3. After n months: $40 + 25n < 33n$ gives $40 < 8n$, $n > 5$. At 5 months both total \$165; from month 6 on the first gym is cheaper (month 6: \$190 versus \$198).
4. $40 + 0.3(240 - 40) = 40 + 60 = 100$. Inverse: $t = (190 - 40)/(240 - 40) = 150/200 = 0.75$.
5. $t = 620/1023$; temperature $-40 + \frac{620}{1023} \cdot 165 = -40 + \frac{102300}{1023} = -40 + 100 = 60$ °C.

5.15. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Graphing calculator

5.16. Mastery check

1. Solve $4(3 - x) + 2x = 5x - 9$.
2. Solve $\dfrac{x - 2}{3} + \dfrac{x + 1}{2} = 4$.
3. Find the solution set of $6x - 2(3x + 1) = -2$. What kind of equation is it?
4. Solve $-2 < 4 - 3x \leq 10$ and write the answer in interval notation.
5. Solve $|2x - 3| \geq 7$.
6. Find the equation of the line through $(1, -2)$ parallel to $2x - y = 5$, and its x-intercept.
7. An animation moves an object from $x = 100$ at frame 0 to $x = 400$ at frame 60. Where is it at frame 45, and at which frame does it first pass $x = 180$?

Decision rule: All correct without hesitation → move on to Chapter 6. Stuck on 1–3 → reread Section 5.2, Section 5.3 and Section 5.4. Stuck on 4–5 → reread Section 5.6 and Section 5.8. Stuck on 6–7 → reread Section 5.10 and Section 5.13. Stuck on more than three → work the section exercises before continuing.

▼ **Mastery check answers**

1. Expand: $12 - 4x + 2x = 5x - 9$, so $12 - 2x = 5x - 9$. Add $2x$ and 9: $21 = 7x$, $x = 3$. Check: left $4(0) + 6 = 6$, right $15 - 9 = 6$.
2. Multiply by 6: $2(x - 2) + 3(x + 1) = 24$, so $2x - 4 + 3x + 3 = 24$, $5x - 1 = 24$, $x = 5$. Check: $\frac{3}{3} + \frac{6}{2} = 1 + 3 = 4$.
3. $6x - 6x - 2 = -2$, i.e. $-2 = -2$: an identity. Solution set $\mathbb{R}$ (all real numbers).
4. Subtract 4: $-6 < -3x \leq 6$. Divide by -3, flipping both: $2 > x \geq -2$, i.e. $-2 \leq x < 2$. Interval $[-2, 2)$.
5. $2x - 3 \geq 7$ gives $x \geq 5$; $2x - 3 \leq -7$ gives $x \leq -2$. Solution $(-\infty, -2] \cup [5, \infty)$.
6. $2x - y = 5$ is $y = 2x - 5$, slope 2. Through $(1, -2)$: $y + 2 = 2(x - 1)$, so $y = 2x - 4$. x-intercept: $2x - 4 = 0$, $x = 2$, the point $(2, 0)$.
7. Frame 45: $t = 45/60 = 0.75$, position $100 + 0.75 \cdot 300 = 325$. For $x = 180$: $t = (180 - 100)/300 = 80/300 \approx 0.267$, frame $0.267 \times 60 = 16$. It reaches 180 exactly at frame 16 and passes it from frame 17 on (frame 17 is at $100 + 300 \cdot \frac{17}{60} = 185$).

US curriculum

Slope, slope–intercept form and graphing lines are core **Algebra I** (grade 8 standards 8.EE.5–6 and 8.F, then high-school A-CED and F-LE). Point–slope and standard form, parallel and perpendicular slopes, and linear modelling from tables and word problems are also Algebra I, revisited in **Geometry** (coordinate proofs) and **Algebra II**. Linear interpolation rarely appears by name in school, though "estimating from a table" does; it is standard in physics, engineering and computer graphics courses.

Chapter 6. Systems of Equations

One equation with two unknowns has infinitely many solutions, so on its own it can't pin anything down. Add a second, independent piece of information and the possibilities usually narrow to one. This chapter is about solving several equations at once. We start with pictures (lines that cross), work through the two classic hand methods (substitution and elimination), and then scale up to Gaussian elimination, the algorithm under every linear solver you'll ever call. We finish with inequalities, feasible regions, and a first look at linear programming. Whenever several facts must hold at once (a phone fixing its position, a circuit simulator, a factory schedule, a model fitted to data) there is a system of equations underneath.

Where this chapter sits

You need: solving one linear equation and the line $y = mx + b$ (Chapter 5), and comfortable manipulation of expressions: expanding, collecting like terms, rearranging (Chapter 4). Fractions show up in the coefficients (Chapter 2).

This chapter unlocks: matrices and their inverses (Volume 2, Chapter 27), which recast everything here in a compact notation; fitting curves through points (Chapter 7); least-squares regression (Volume 3, Chapter 31); conditioning and floating-point stability (Volume 2, Chapter 32); and optimisation in general (Volume 2, Chapter 18 , Volume 2, Chapter 20).

6.1. Why one equation is not enough

6.1.1. The problem

A school sells 50 tickets to a concert. Adult tickets cost \$12 and child tickets cost \$7. The till holds \$465. How many of each were sold?

Call the number of adult tickets a and the number of child tickets c. The money gives you one equation:

$$12a + 7c = 465.$$

That's a perfectly good linear equation, but it can't answer the question on its own. Plenty of pairs satisfy it if you ignore the head count: $a = 0, c = 66.43\ldots$ is one, $a = 38.75, c = 0$ is another, and there are infinitely many in between. The ticket count gives you a **second** fact:

$$a + c = 50.$$

Taken alone, this one is also satisfied by infinitely many pairs: (0, 50), (1, 49), (25, 25), and so on. But very few pairs satisfy **both** equations together. In fact exactly one does, and finding it is the subject of this chapter. (It's $a = 23,\ c = 27$. Check it: $23 + 27 = 50$ and $12 \cdot 23 + 7 \cdot 27 = 276 + 189 = 465$.)

6.1.2. Intuition: each equation is a constraint

Think of each equation as a filter. Before any information arrives, every pair (a, c) is possible. Each equation throws away everything that doesn't satisfy it. With one equation in two unknowns you keep a whole line of candidates. A second equation keeps another line, and what survives both filters is where the two lines cross.

This gives a rule of thumb you'll meet again and again:

> *To pin down n unknowns you usually need n independent equations.*

"Usually" and "independent" are doing real work in that sentence. Two equations that say the same thing in different words ($a + c = 50$ and $2a + 2c = 100$) count as only one piece of information. Two equations that contradict each other can't both hold, so nothing survives. We'll be precise about both cases in Section 6.5.



System of equations

A **system of equations** is a collection of equations that must all hold **at the same time**, for the same values of the unknowns. A brace on the left is the usual way to show they belong together:

$$\begin{cases} a + c = 50 \\ 12a + 7c = 465 \end{cases}$$

British courses call these **simultaneous equations**, a name that stresses the "at the same time" part. US courses say **systems of equations**. They mean the same thing.



Solution of a system

A **solution** of a system is an assignment of values to **all** the unknowns that makes **every** equation true. For two unknowns x, y a solution is an **ordered pair** (x, y); for three it's an **ordered triple** (x, y, z). The **solution set** is the set of all solutions. It may contain exactly one element, none, or infinitely many.



Linear equation (in several variables)

An equation is **linear** if each unknown appears only to the first power, multiplied by a constant, and unknowns are never multiplied together. Its **standard form** is

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b.$$

Here the subscripted $x_1, x_2, \ldots, x_n$ are the unknowns, the a_i are their **coefficients**, and b is the **constant term** (or **right-hand side**). So $3x - 2y = 7$ is linear. $xy = 7$, $x^2 + y = 1$ and $\sqrt{x} = y$ are not.

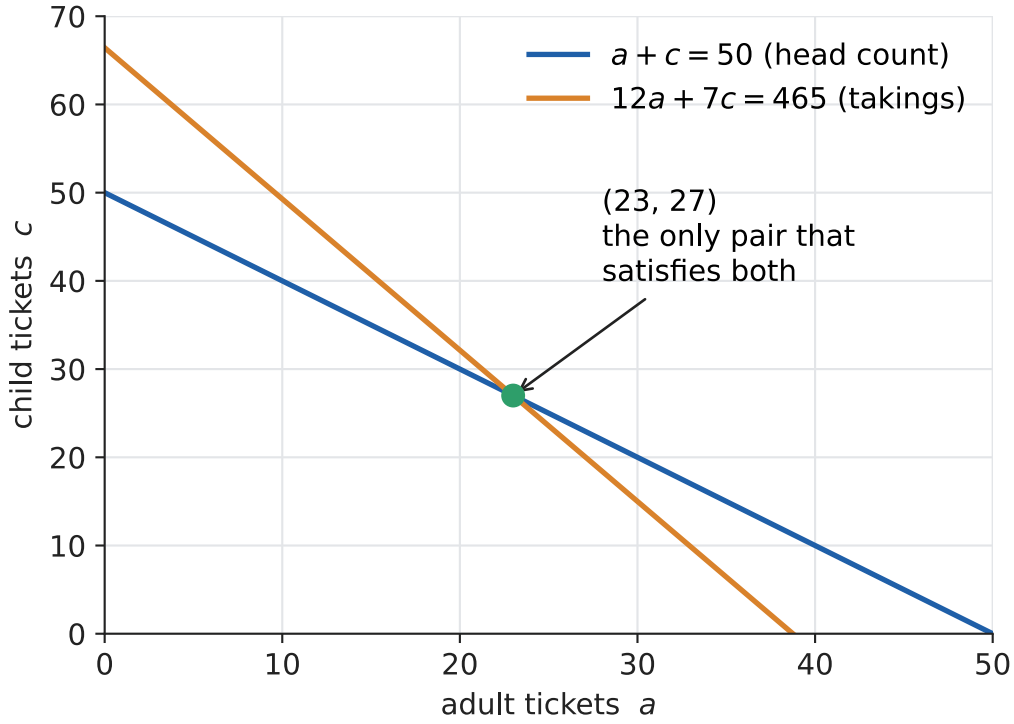


Figure 39. The ticket problem as two lines. Each line holds every pair that satisfies one equation; only the crossing point satisfies both.



Nine chapters, two thousand years

Systems of linear equations are among the oldest problems in written mathematics. The Chinese text *The Nine Chapters on the Mathematical Art* (compiled around the 1st century CE) has a chapter called *fangcheng* ("rectangular arrays"). It solves systems of three or more equations by laying out the coefficients on a counting board in columns and subtracting multiples of one column from another. That is Gaussian elimination, about 1,700 years before Gauss. The board even had a way to represent negative numbers: red rods for positive, black for negative.

6.2. Pictures first: the graphical method

6.2.1. Why a picture works

From Chapter 5 you know that the solutions of one linear equation in x and y form a straight line in the plane. A point lies on the line exactly when its coordinates satisfy the equation. So a point that satisfies **two** equations must lie on **both** lines at once, which means it's an intersection point.

> *Solving a system of two linear equations = finding where two lines meet.*

Two straight lines in a plane can relate in only three ways:

- They **cross** at exactly one point, giving **one solution**.
- They're **parallel** (same slope, different intercept) and never meet, giving **no solution**.

- They're **the same line** drawn twice, so every point on it is shared, giving **infinitely many solutions**.

Nothing else can happen. Two distinct straight lines can't meet at exactly two points. That fact matters: if you ever find two different solutions to a linear system, there must be infinitely many.

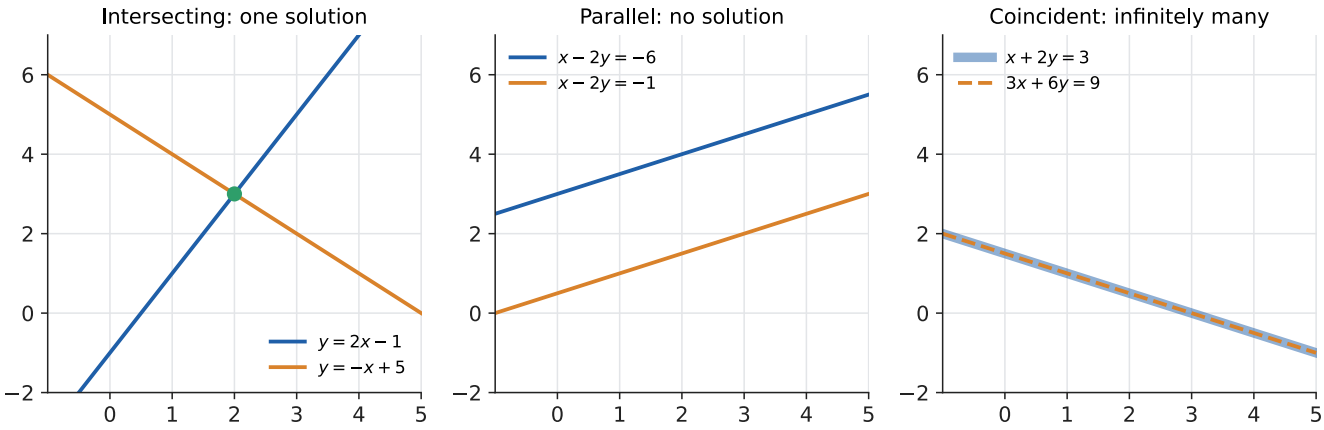


Figure 40. The three possible relationships between two lines, and the three possible sizes of the solution set.



Consistent, inconsistent, independent, dependent

- A system is **consistent** if it has at least one solution, and **inconsistent** if it has none (parallel lines).
- A consistent system is **independent** if the solution is unique (crossing lines), and **dependent** if there are infinitely many solutions (coincident lines). The second equation then "depends on" the first: it's a multiple of it and adds no new information.

US textbooks use all four words. A common summary is "consistent independent / consistent dependent / inconsistent".

6.2.2. Worked example: solving by graphing

Solve

$$\begin{cases} y = 2x - 1 \\ y = -x + 5 \end{cases}$$

Both equations are already in slope–intercept form, $y = mx + b$.

1. **First line.** Slope 2, intercept -1. Plot $(0, -1)$, then go 1 right and 2 up to $(1, 1)$, then again to $(2, 3)$.
2. **Second line.** Slope -1, intercept 5. Plot $(0, 5)$, then 1 right and 1 down to $(1, 4)$, then $(2, 3)$.
3. **Read off the crossing.** Both lines pass through $(2, 3)$ (see the left panel of the figure above).
4. **Check algebraically.** First equation: $2(2) - 1 = 3$. Second: $-(2) + 5 = 3$. Both hold.

Solution: $(x, y) = (2, 3)$.

6.2.3. Predicting the case before you solve

Put both equations in slope–intercept form and compare:

Slopes	Intercepts	Result
different	anything	one solution (lines cross)
equal	different	no solution (parallel)
equal	equal	infinitely many (same line)

For example, $x - 2y = -6$ rearranges to $y = \frac{1}{2}x + 3$, and $x - 2y = -1$ to $y = \frac{1}{2}x + \frac{1}{2}$. Same slope, different intercepts: parallel, so no solution. That's the middle panel.

6.2.4. Failure modes of graphing



Graphs are approximate

Graphing is excellent for **seeing** what kind of answer to expect and terrible for **precise** answers. If the true solution is $(\frac{17}{11}, \frac{30}{11}) \approx (1.545, 2.727)$, no hand-drawn graph will tell you that. And when two lines are nearly parallel, a tiny drawing error moves the apparent crossing a long way (we'll see in Section 6.11 that computers have a version of this problem too). Use graphs to sanity-check and to understand. Use algebra to get exact answers.

Graphing is still worth doing in three situations: when you want a quick estimate; when you need to understand **what kind** of answer exists before computing it; and when the equations aren't linear and no algebraic method is convenient. Finding where $y = 2^x$ meets $y = x + 3$ is a graphing (or numerical) job, not a substitution job.



Drag the lines

Show two lines with sliders for each slope and intercept, and display the intersection point and its coordinates live. Rotate one line until it's parallel to the other and watch the intersection run off to infinity and vanish. Then slide its intercept until the lines coincide. A second mode shows the equations in standard form $ax + by = c$ with sliders on a, b, c, so the learner sees how coefficients control slope.

Try it: Two lines with sliders

6.3. Substitution

6.3.1. The idea

If one equation tells you what one unknown **equals** in terms of the other, you can replace that unknown everywhere in the other equation. This is the "equals can be substituted for equals" principle from Chapter 4. What's left is a single equation in a single unknown, which you already know how to solve (Chapter 5).

Why is this legitimate? Suppose (x, y) is a solution of the system. Then x really does equal $y + 2$ (say), so writing $y + 2$ wherever x appeared changes nothing about whether the other equation is true. Every solution of the original system is also a solution of the reduced equation. Then, by back-substituting, every solution of the reduced equation gives back a solution of the system. Nothing is gained and nothing is lost.

Example 35. Procedure — Solving a 2×2 system by substitution

Input: two linear equations in unknowns x and y.

Steps:

1. Choose an equation and an unknown that's easy to isolate. Ideally pick one with coefficient 1 or -1.
2. Rearrange that equation to make the chosen unknown the subject, e.g. $x = (\text{expression in } y)$.
3. Substitute that expression, **in brackets**, into the **other** equation.
4. Solve the resulting one-variable equation.
5. Substitute the value back into the rearranged equation from step 2 to get the other unknown.
6. Check the pair in **both** original equations.

Output: the solution (x, y); or, if step 4 collapses to a false statement like $0 = 5$ or a true one like $0 = 0$, the conclusion "no solution" or "infinitely many" (see Section 6.5).

6.3.2. Worked example

Solve

$$\begin{cases} 3x + 2y = 16 & (1) \\ x - y = 2 & (2) \end{cases}$$

Equation (2) has x with coefficient 1, so isolate x there:

$$\begin{aligned} x - y &= 2 \\ x &= y + 2 \qquad \text{(add } y \text{ to both sides)} \quad (3) \end{aligned}$$

Substitute (3) into (1), keeping the brackets:

$$
\begin{aligned}
3(y+2)+2y &= 16 && \text{(replace } x \text{ by } y+2\text{)}\\
3y+6+2y &= 16 && \text{(distribute the 3)}\\
5y+6 &= 16 && \text{(collect like terms)}\\
5y &= 10 && \text{(subtract 6)}\\
y &= 2 && \text{(divide by 5)}
\end{aligned}
$$

Back-substitute into (3): $x = 2 + 2 = 4$.

Check. (1): $3(4) + 2(2) = 12 + 4 = 16$. (2): $4 - 2 = 2$. Both hold, so $(x, y) = (4, 2)$.

6.3.3. The ticket problem by substitution

From $a + c = 50$ we get $c = 50 - a$. Substitute into $12a + 7c = 465$:

$$
\begin{aligned}
12a + 7(50 - a) &= 465 && \\
12a + 350 - 7a &= 465 && \text{(distribute)}\\
5a + 350 &= 465 && \text{(collect)}\\
5a &= 115 && \\
a &= 23 &&
\end{aligned}
$$

Then $c = 50 - 23 = 27$. That's 23 adult tickets and 27 child tickets.



Three classic substitution slips

1. **Dropping the brackets.** Substituting $x = y + 2$ into $3x$ gives $3(y + 2) = 3y + 6$, not $3y + 2$. Always write the brackets first, then expand.
2. **Substituting back into the same equation.** If you rearranged (2) to get (3) and then substitute (3) into (2), you get $y + 2 - y = 2$, i.e. $2 = 2$. That's true but tells you nothing, because you've used the same information twice. Substitute into the **other** equation.
3. **Stopping at one unknown.** The question asked for both. Finish with the back-substitution and the check.

Substitution shines when one equation already gives a variable explicitly ($y = \ldots$) or has a coefficient of ± 1. It's also the method of choice when one equation is **not** linear. For example, with $y = x^2$ and $y = x + 2$, substitution gives $x^2 = x + 2$, a quadratic you'll learn to solve in Chapter 8. When every coefficient is something like 7 or 13, substitution drags fractions in early, and elimination is cleaner.

6.4. Elimination

6.4.1. The problem with substitution

Try substitution on

$$
\begin{cases}
2x + 3y = 12 & (1)\\
5x - 4y = 7 & (2)
\end{cases}
$$

Isolating anything gives fractions straight away: $x = 6 - \frac{3}{2}y$. It works, but it's fiddly. Elimination avoids this.

6.4.2. Why adding equations is legal

If $A = B$ and $C = D$, then $A + C = B + D$: add equal things to equal things and the results are equal. Likewise, $kA = kB$ for any number k. So from any two equations in a system you can form a **new** true equation by scaling each one and adding them. The trick is to choose the scale factors so that one unknown **cancels**. Its coefficients become opposites and add to zero, which **eliminates** it.

There's a subtle point: replacing one equation by "itself plus a multiple of the other" gives a new system with **exactly the same solutions** (as long as the multiplier on the equation being replaced is nonzero). The operation can be undone: just subtract the same multiple again. Reversible steps can't create or destroy solutions. That one observation is the whole correctness argument for elimination, and later for Gaussian elimination.



Equivalent systems

Two systems are **equivalent** if they have exactly the same solution set. Elimination and substitution are chains of equivalent systems, each simpler than the one before, ending in a system so simple you can read the answer off (like $x = 3,\ y = 2$).

Example 36. Procedure — Solving a 2×2 system by elimination

Input: two linear equations in standard form $ax + by = c$, with like terms lined up in columns.

Steps:

1. Choose the unknown to eliminate. Prefer one whose coefficients are already equal or opposite, or have a small common multiple.
2. Multiply each equation by a nonzero number so that the chosen unknown has coefficients that are **opposites** (or **equal**). The least common multiple of the two coefficients keeps the numbers small.
3. Add the equations if the coefficients are opposites, or subtract if they're equal. The chosen unknown disappears.
4. Solve the resulting one-variable equation.
5. Substitute the value into either original equation and solve for the other unknown.
6. Check in **both** original equations.

Output: the solution (x, y), or the detection of "no solution" (a false statement such as $0 = -4$) or "infinitely many" ($0 = 0$).

6.4.3. Worked example

Solve $2x + 3y = 12$ (1) and $5x - 4y = 7$ (2).

Choose. The y-coefficients are 3 and -4. They already have opposite signs, and their least common multiple is 12. So scale (1) by 4 and (2) by 3:

$$\begin{aligned} 4\times(1):&\quad 8x+12y=48\\ 3\times(2):&\quad 15x-12y=21 \end{aligned}$$

Add (the y-terms are opposites):

$$\begin{aligned} (8x+15x)+(12y-12y)&=48+21\\ 23x&=69\\ x&=3 \qquad\qquad (\text{divide by } 23) \end{aligned}$$

Back-substitute into (1):

$$\begin{aligned} 2(3)+3y&=12\\ 6+3y&=12\\ 3y&=6\\ y&=2 \end{aligned}$$

Check in (2): $5(3)-4(2)=15-8=7$. The solution is $(x,y)=(3,2)$.



Subtracting a negative

When you **subtract** one equation from another, the subtraction applies to **every** term, including the right-hand side and any terms that are already negative. From

$$\begin{aligned} 6x-15y&=-72\\ 6x+8y&=20 \end{aligned}$$

subtracting the second from the first gives $-15y-8y=-72-20$, i.e. $-23y=-92$, so $y=4$. A very common slip is to write $-15y+8y$. If you find sign handling error-prone, multiply one equation by a negative number instead so you can always **add**. Adding is harder to get wrong.

6.4.4. Alternative approach: the general formula (Cramer's rule)

Elimination is a procedure, so we can run it once on **symbolic** coefficients and get a formula that works for every 2×2 system. Take

$$\begin{cases} ax+by=c & (1)\\ dx+ey=f & (2) \end{cases}$$

To eliminate y, multiply (1) by e and (2) by b, then subtract:

$$\begin{aligned} aex+bey&=ce && (e\times(1))\\ bdx+bey&=bf && (b\times(2))\\ (ae-bd)\,x&=ce-bf && (\text{subtract}) \end{aligned}$$

To eliminate x, multiply (1) by d and (2) by a, then subtract the first from the second:

$$
\begin{aligned}
adx + bdy &= cd && (d \times (1)) \\
adx + aey &= af && (a \times (2)) \\
(ae - bd)\,y &= af - cd && (\text{subtract})
\end{aligned}
$$

So, provided $ae - bd \neq 0$,

$$
x = \frac{ce - bf}{ae - bd}, \qquad y = \frac{af - cd}{ae - bd}.
$$

The quantity $ae - bd$ is called the **determinant** of the coefficient array. It's zero exactly when the two lines have the same slope, that is, when there isn't a unique solution. You'll meet determinants properly in Volume 2, Chapter 27 , where this formula becomes **Cramer's rule**. Here it is as code:

```python
def solve_2x2(a, b, c, d, e, f):
    """Solve  a*x + b*y = c,  d*x + e*y = f  by elimination."""
    det = a * e - b * d          # the "cross-multiplied" leftover coefficient
    if det == 0:
        raise ValueError("lines are parallel or coincident: no unique solution")
    x = (c * e - b * f) / det
    y = (a * f - c * d) / det
    return x, y

print(solve_2x2(2, 3, 12, 5, -4, 7))      # worked example
print(solve_2x2(1, 1, 50, 12, 7, 465))    # tickets
try:
    solve_2x2(2, 4, 6, 1, 2, 5)
except ValueError as err:
    print("error:", err)
```

```
(3.0, 2.0)
(23.0, 27.0)
error: lines are parallel or coincident: no unique solution
```

Should you memorise this formula? For 2×2 systems it's handy, and it's what you'd put in a tight inner loop (for example, intersecting two line segments in a game engine). But it doesn't scale: the analogous formula for n unknowns is hopelessly expensive. What scales is the **procedure**, which is why we come back to it in Section 6.9.

6.5. When the algebra says "never" or "always"

Graphs showed three possible outcomes. Here's how each one looks when you do the algebra. You don't need to graph anything: the algebra announces the case itself.

6.5.1. No solution: a contradiction appears

Solve

$$
\begin{cases}
2x + 4y = 6 & (1) \\
x + 2y = 5 & (2)
\end{cases}
$$

Multiply (2) by 2 and subtract it from (1):

$$\begin{aligned} 2x + 4y &= 6 \\ 2x + 4y &= 10 \qquad (2 \times (2)) \\ 0 &= -4 \qquad (\textbf{subtract}) \end{aligned}$$

Both unknowns vanished and what's left is false. That isn't an arithmetic error. It's the system telling you that no pair (x, y) can satisfy both equations. Geometrically, $2x + 4y = 6$ and $x + 2y = 5$ are the parallel lines $y = -\frac{1}{2}x + \frac{3}{2}$ and $y = -\frac{1}{2}x + \frac{5}{2}$. The system is **inconsistent**, and the solution set is the empty set, written $\varnothing$.

The logic is a proof by contradiction (Volume 3, Chapter 22). **If** a solution existed, the derived equation $0 = -4$ would be true for it. Since $0 = -4$ is never true, no solution exists.

6.5.2. Infinitely many solutions: an identity appears

Solve

$$\begin{cases} x + 2y = 3 & (1) \\ 3x + 6y = 9 & (2) \end{cases}$$

Multiply (1) by 3 and subtract from (2):

$$\begin{aligned} 3x + 6y &= 9 \\ 3x + 6y &= 9 \qquad (3 \times (1)) \\ 0 &= 0 \qquad (\textbf{subtract}) \end{aligned}$$

Now both unknowns vanish and what's left is **always** true. Equation (2) is just three times equation (1), so it carries no new information. You really have one equation in two unknowns, and every point on the line $x + 2y = 3$ is a solution. The system is **consistent and dependent**.

How do you **write down** infinitely many solutions? Let one unknown be free and express the other in terms of it. Introduce a **parameter** t (any real number) and set $y = t$. Then $x = 3 - 2t$:

$$(x, y) = (3 - 2t,\ t), \qquad t \in \mathbb{R}.$$

Here $t \in \mathbb{R}$ reads "t is an element of the real numbers", i.e. t can be any real number. Every value of t produces one solution: $t = 0$ gives $(3, 0)$, $t = 1$ gives $(1, 1)$, $t = -2$ gives $(7, -2)$. You can check each one in both equations.



Parameter, parametric solution

A **parameter** is a letter that stands for an arbitrary value you're free to choose, as opposed to an unknown you're trying to find (compare Chapter 4). A **parametric solution** such as $(3 - 2t,\ t)$ describes an infinite solution set by saying how every solution is built from the parameter. The unknown you set equal to the parameter is called a **free variable**.

6.5.3. A quick test without solving

For $ax + by = c$ and $dx + ey = f$, compare the ratios of the coefficients:

- $\frac{a}{d} \neq \frac{b}{e}$ (equivalently $ae - bd \neq 0$): exactly one solution.
- $\frac{a}{d} = \frac{b}{e} \neq \frac{c}{f}$: no solution.
- $\frac{a}{d} = \frac{b}{e} = \frac{c}{f}$: infinitely many.

(If any of d, e, f is zero, use the cross-multiplied form, e.g. $ae = bd$, to avoid dividing by zero.)



0 = 0 does not mean x = 0

Students sometimes see $0 = 0$ and write "$x = 0$", or see $0 = -4$ and assume they slipped somewhere and start again. Learn to read these as **messages**: $0 = 0$ means "the equations were redundant"; $0 = \textbf{nonzero}$ means "the equations contradict each other". Do re-check your arithmetic once, because an arithmetic error **can** produce either, but then believe the result.

Computer-science connection: degenerate cases are where code breaks

The parallel and coincident cases are exactly the ones naive geometry code forgets. A function that intersects two line segments by computing $(ce - bf)/(ae - bd)$ will divide by zero when the segments are parallel, and for **nearly** parallel segments it will return a point miles away. Robust computational-geometry libraries handle "determinant exactly zero" as a special case, and "determinant tiny" as a separate, suspicious case, often using exact or adaptive-precision arithmetic for the sign test. When you write `if det == 0` with floating-point numbers, you're nearly always writing a bug; compare against a tolerance scaled to the size of the inputs instead. We return to this in Volume 2, Chapter 32 .

6.6. Choosing a method for two equations

All three methods give the same answer when they work. They differ in effort, in error-proneness, and in what they tell you.

Method	Choose it when	Avoid it when
Graphing	You want to see the situation, estimate, or check the number of solutions; equations aren't linear and nothing else is easy	You need an exact answer, or the lines are nearly parallel
Substitution	A variable already has coefficient ± 1, or an equation is given as $y = \ldots$; one equation is nonlinear	Every coefficient is "awkward" (fractions appear at once)
Elimination	Equations are in standard form $ax + by = c$; coefficients are integers; you'll scale up to 3+ unknowns	An equation is nonlinear (you can't cancel x^2 against x)
Formula ($\det$)	Coding a 2×2 solver; symbolic coefficients	Hand-working larger systems

A practical habit: before choosing, **clear fractions and decimals** by multiplying each equation by a suitable number (Chapter 5). The system

$$\begin{cases} \frac{x}{2} + \frac{y}{3} = 4 \\ 0.3x - 0.2y = 0 \end{cases}$$

becomes $3x + 2y = 24$ (times 6) and $3x - 2y = 0$ (times 10), and adding them gives $6x = 24$ at once.

Another habit: look for **symmetry**. With $3c + 2m = 14.5$ and $2c + 3m = 13$, adding gives $5c + 5m = 27.5$, so $c + m = 5.5$. Subtracting gives $c - m = 1.5$. Two trivial equations replace two awkward ones.

6.6.1. Where you'd use this

Whenever two facts constrain two unknowns, you have a 2×2 system.

- **Sport.** A basketball team scores 52 points from 23 field goals, all 2-pointers or 3-pointers. With x twos and y threes: $x + y = 23$ and $2x + 3y = 52$. Subtract twice the first from the second: $y = 6$, so $x = 17$. Check: $34 + 18 = 52$.
- **Economics.** A market settles where supply meets demand. If suppliers offer quantity q at price $p = 2 + 0.5q$ and buyers demand it at $p = 20 - q$, then $2 + 0.5q = 20 - q$ gives $q = 12$ and $p = 8$: the **equilibrium** is the crossing point of two lines.
- **Positioning.** Your phone can locate itself from distances to known transmitters (GPS satellites, Wi-Fi access points). In a flat 2D version, beacons sit at $(0, 0)$, $(10, 0)$ and $(0, 10)$, and the measured squared distances are 25, 65 and 45. Each distance gives a circle equation such as $x^2 + y^2 = 25$ and $(x - 10)^2 + y^2 = 65$. Subtracting the first from the second cancels x^2 and y^2, leaving the

linear equation $-20x + 100 = 40$, so $x = 3$. The third beacon gives $-20y + 100 = 20$ in the same way, so $y = 4$. The position is $(3, 4)$. Real receivers do this in 3D with a fourth unknown (the clock error), and with more measurements than unknowns, solved by least squares (Section 6.11).

6.7. Exercises: two equations, two unknowns

1. **(Recognition)** Without solving, say whether each system has one solution, no solution, or infinitely many.

   a. $y = 3x + 1$, $y = 3x - 4$

   b. $2x - y = 4$, $6x - 3y = 12$

   c. $x + y = 1$, $x - y = 1$

2. **(Substitution)** Solve $y = 4x - 3$, $2x + 3y = 19$.

3. **(Elimination)** Solve $3x + 4y = 10$, $2x - 5y = -24$.

4. **(Fractions)** Solve $\frac{x}{2} + \frac{y}{3} = 4$, $\frac{x}{4} - \frac{y}{6} = 0$.

5. **(Word problem)** Three coffees and two muffins cost $14.50. Two coffees and three muffins cost $13.00. Find the price of each.

6. **(Parameter)** For which value of k does $kx + 2y = 5$, $3x + 6y = 1$ have no solution? Is there a value of k that gives infinitely many?

7. **(Modelling)** A boat travels 24 km downstream in 2 hours and returns upstream in 3 hours. Find the boat's speed in still water and the speed of the current.

▼ **Answers**

1. (a) Same slope 3, different intercepts: **no solution**. (b) The second equation is 3 times the first: **infinitely many**. (c) Slopes -1 and 1 differ: **one solution**, namely $(1, 0)$.

2. Substitute: $2x + 3(4x - 3) = 19 \Rightarrow 2x + 12x - 9 = 19 \Rightarrow 14x = 28 \Rightarrow x = 2$. Then $y = 4(2) - 3 = 5$. Check: $4 + 15 = 19$. Answer: $(2, 5)$.

3. Multiply the first by 2 and the second by 3: $6x + 8y = 20$, $6x - 15y = -72$. Subtract (first minus second): $23y = 92$, so $y = 4$. Then $3x + 16 = 10 \Rightarrow x = -2$. Check: $2(-2) - 5(4) = -24$. Answer: $(-2, 4)$.

4. Multiply the first by 6: $3x + 2y = 24$. Multiply the second by 12: $3x - 2y = 0$. Add: $6x = 24$, $x = 4$; then $2y = 3x = 12$, $y = 6$. Check: $2 + 2 = 4$ and $1 - 1 = 0$. Answer: $(4, 6)$.

5. $3c + 2m = 14.5$ and $2c + 3m = 13$. Add: $5c + 5m = 27.5 \Rightarrow c + m = 5.5$. Subtract: $c - m = 1.5$. Adding these: $2c = 7$, so $c = 3.50$ and $m = 2.00$. Coffee $3.50, muffin $2.00.

6. No solution needs $ae = bd$, i.e. $6k = 2 \cdot 3$, so $k = 1$. Then the equations are $x + 2y = 5$ and $x + 2y = \frac{1}{3}$, which contradict each other, so there's no solution. Infinitely many would need $\frac{k}{3} = \frac{2}{6} = \frac{5}{1}$, but $\frac{2}{6} \neq 5$, so no k works.

7. Let b be the boat speed and c the current, both in km/h. Downstream speed is $24/2 = 12$, so $b + c = 12$. Upstream speed is $24/3 = 8$, so $b - c = 8$. Adding: $2b = 20$, $b = 10$, so $c = 2$. Boat 10 km/h, current 2 km/h.

US curriculum

Systems of two linear equations (graphing, substitution, elimination, and the "no solution / infinitely many" cases) are a core **Algebra I** topic, usually in grade 8 or 9. Common Core standards **8.EE.C.8** and **A-REI.C.5–6** cover it. A-REI.C.5 explicitly asks students to justify **why** replacing an equation by itself plus a multiple of the other preserves the solutions, which is the argument in Section 6.4. The UK name "simultaneous equations" is unknown to most US students.

6.8. Three equations, three unknowns

6.8.1. The problem

A café sells three blends of coffee bean, and you want to know how many kilograms of each to roast. Or a physicist has three currents in a circuit. Or you want the parabola $y = ax^2 + bx + c$ through three given points (you'll do this in the exercises). Three unknowns need three independent equations, and the methods above extend directly.

6.8.2. What it looks like

A linear equation in three unknowns, such as $x + y + z = 6$, describes a flat **plane** in three-dimensional space. Three planes can meet in a single point (the typical case, shown below), or along a whole line (like the pages of a book meeting at the spine), or coincide entirely, or have no point common to all three. That last case covers parallel planes, but also three planes that meet pairwise in three parallel lines, like the faces of a triangular prism. So the menu of outcomes is the same as before: one solution, none, or infinitely many.

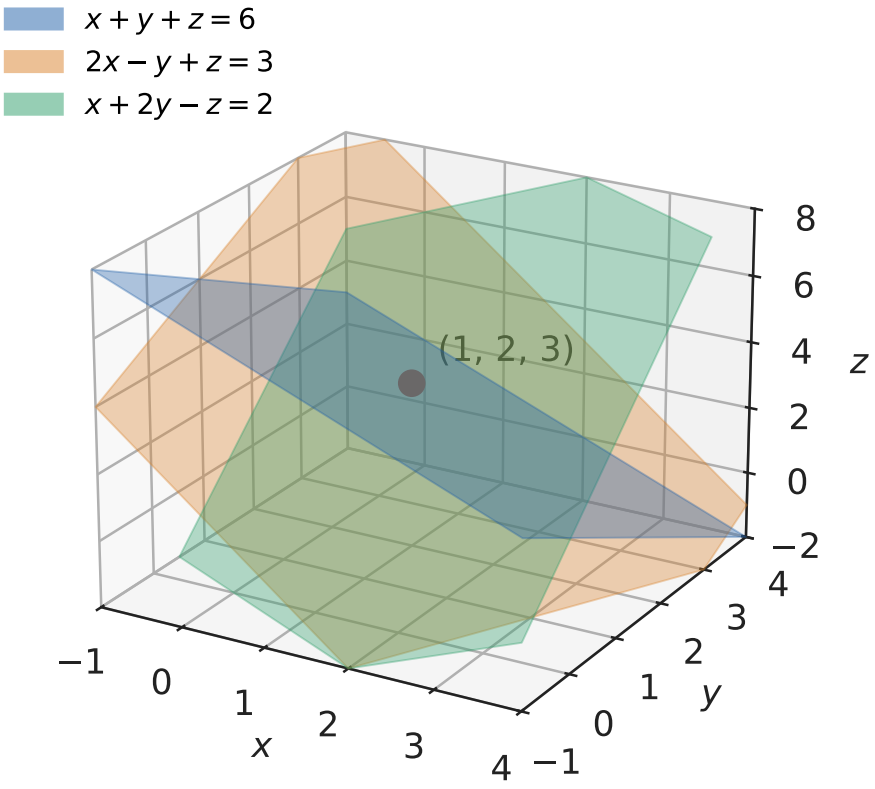


Figure 41. Three planes meeting at a single point. The solution of the system below is where all three intersect.

6.8.3. The method: eliminate the same variable twice

The strategy reduces the problem to one you've already solved:

1. Use two different pairs of equations to eliminate the **same** unknown. That leaves two equations in two unknowns.
2. Solve that 2×2 system.
3. Back-substitute to find the third unknown.

6.8.4. Worked example

Solve

$$\begin{cases} x + y + z = 6 & (1) \\ 2x - y + z = 3 & (2) \\ x + 2y - z = 2 & (3) \end{cases}$$

Eliminate x using (1) and (2). Subtract 2 × (1) from (2):

$$\begin{aligned} (2x - y + z) - 2(x + y + z) &= 3 - 2(6) \\ 2x - y + z - 2x - 2y - 2z &= 3 - 12 \\ -3y - z &= -9 \qquad (4) \end{aligned}$$

Eliminate x again, using (1) and (3). Subtract (1) from (3):

$$(x + 2y - z) - (x + y + z) = 2 - 6$$
$$y - 2z = -4 \qquad (5)$$

Solve the 2×2 system (4), (5). Equation (5) has y with coefficient 1, so either method works. Eliminate y by adding 3 × (5) to (4):

$$(-3y - z) + 3(y - 2z) = -9 + 3(-4)$$
$$-3y - z + 3y - 6z = -9 - 12$$
$$-7z = -21$$
$$z = 3$$

Back-substitute. From (5): $y = -4 + 2z = -4 + 6 = 2$. From (1): $x = 6 - y - z = 6 - 2 - 3 = 1$.

Check in all three originals. (1): $1 + 2 + 3 = 6$. (2): $2 - 2 + 3 = 3$. (3): $1 + 4 - 3 = 2$. The solution is $(x, y, z) = (1, 2, 3)$.



Eliminating different variables

The most common mistake with three unknowns is eliminating x from one pair and y from another. You end up with two equations that each still contain **two** unknowns, but not the same two, and you've made no progress. Decide on the variable first, then eliminate **that** variable twice.

The second most common mistake is using the same pair twice, e.g. (1) & (2) and then (2) & (1) again. Every original equation must take part, or you've thrown away information.

Notice the shape of what we did. We kept (1) as it was, replaced (2) and (3) by equations without x, then replaced one of those by an equation without y. The final system

$$\begin{cases} x + y + z = 6 \\ \quad y - 2z = -4 \\ \quad\quad -7z = -21 \end{cases}$$

is a **triangle**: the last equation has one unknown, the one above it two, the one above that three. Solving a triangle from the bottom up is trivial. Making the triangle is the hard part. Turning this observation into an algorithm is the next section.

6.8.5. Where you'd use this

- **Fitting a model to measurements.** A ball's height is recorded as 2 m at $t = 0$, 12.1 m at $t = 1$ s and 12.4 m at $t = 2$ s. If $h = at^2 + bt + c$, each reading is one linear equation in a, b, c: $c = 2$, $a + b + c = 12.1$, $4a + 2b + c = 12.4$. With $c = 2$, these become $a + b = 10.1$ and $4a + 2b = 10.4$; subtract twice the first from the second to get $2a = -9.8$. So $a = -4.9$, $b = 15$: the data recover $h = 2 + 15t - 4.9t^2$, with $a = -\frac{1}{2}g$ for $g = 9.8\ \mathrm{m/s^2}$. This is the thrown ball of Chapter 8.

- **Chemistry.** Balancing a reaction is a linear system with one equation per element. For propane burning, $\mathrm{C_3H_8} + a\,\mathrm{O_2} \to b\,\mathrm{CO_2} + c\,\mathrm{H_2O}$: carbon gives $b = 3$, hydrogen $2c = 8$, oxygen $2a = 2b + c$. The system is already triangular: $b = 3$, $c = 4$, $a = 5$. Larger reactions give larger systems, and software balances them with exactly the elimination of the next section.
- **Colour.** A screen colour is a mix of red, green and blue light, and converting to another colour space (for example, the device-independent XYZ space used in colour management) is three linear equations in three unknowns. Converting back means solving them.

6.9. Gaussian elimination: the algorithm

6.9.1. The problem

A structural-engineering model might have 10,000 unknowns, a circuit simulator a million, a weather model far more. Nobody juggles named equations like "(4)" and "(5)" at that scale. We need a procedure that's mechanical, works for any number of unknowns, detects the no-solution and infinitely-many cases automatically, and can be coded in a few lines. That's **Gaussian elimination**.

6.9.2. Notation: the augmented matrix

Notice that the letters x, y, z did no work in the elimination. They only marked which **column** each coefficient lived in. So drop them and keep just the numbers in a rectangular array:

$$\begin{cases} x + y + z = 6 \\ 2x - y + z = 3 \\ x + 2y - z = 2 \end{cases} \quad\longleftrightarrow\quad \left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 2 & -1 & 1 & 3 \\ 1 & 2 & -1 & 2 \end{array}\right]$$

Each **row** is one equation. Each of the first three **columns** holds the coefficients of one unknown, and the column after the vertical bar holds the right-hand sides.



Matrix, augmented matrix, row echelon form, pivot

- A **matrix** is a rectangular array of numbers arranged in **rows** (horizontal) and **columns** (vertical). An $m \times n$ matrix has m rows and n columns. Matrices get a full chapter of their own in Volume 2, Chapter 27 .
- The **coefficient matrix** of a linear system holds just the coefficients. The **augmented matrix** has the right-hand-side column attached ("augmented") after a vertical bar.
- A matrix is in **row echelon form** when every row's first nonzero entry is to the right of the first nonzero entry of the row above, and any all-zero rows are at the bottom. It's a staircase ("échelon" is French for a rung of a ladder).
- The first nonzero entry in each row of an echelon form is a **pivot**. The pivot is the entry you use to clear the entries beneath it.

6.9.3. Why it works: the three row operations

Everything we did by hand used only these three moves, each of which turns the system into an **equivalent** one (same solutions) because each can be undone:

1. **Swap** two rows. (Reorder the equations. Undo: swap back.)
2. **Scale** a row by a nonzero number. (Undo: divide by it.)
3. **Replace** a row by itself plus a multiple of another row. (Undo: subtract the same multiple.)

Notation: $R_2 \leftarrow R_2 - 2R_1$ means "replace row 2 by row 2 minus twice row 1". The arrow $\leftarrow$ is deliberately the programmer's assignment, not an equation.

6.9.4. Worked example in matrix form

Here is the same system again, in matrix form.

Step 1: clear column 1 below the pivot 1 in row 1.

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 2 & -1 & 1 & 3 \\ 1 & 2 & -1 & 2 \end{array}\right] \xrightarrow[R_3 \leftarrow R_3 - R_1]{R_2 \leftarrow R_2 - 2R_1} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & -3 & -1 & -9 \\ 0 & 1 & -2 & -4 \end{array}\right]$$

Row 2: $(2,-1,1,3) - 2(1,1,1,6) = (0,-3,-1,-9)$. Row 3: $(1,2,-1,2) - (1,1,1,6) = (0,1,-2,-4)$.

Step 2: choose a pivot for column 2. We could use -3, but then we'd have to add $\frac{1}{3}$ of row 2 to row 3, which brings in fractions. Swapping rows 2 and 3 gives a pivot of 1 instead:

$$\xrightarrow{R_2 \leftrightarrow R_3} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & -2 & -4 \\ 0 & -3 & -1 & -9 \end{array}\right]$$

Step 3: clear column 2 below the pivot.

$$\xrightarrow{R_3 \leftarrow R_3 + 3R_2} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & -2 & -4 \\ 0 & 0 & -7 & -21 \end{array}\right]$$

Row 3: $(0,-3,-1,-9) + 3(0,1,-2,-4) = (0,0,-7,-21)$. This is row echelon form.

Step 4: back substitution. Read the rows as equations, bottom up:

$$\begin{aligned} -7z &= -21 &\Rightarrow\quad z &= 3 \\ y - 2z &= -4 &\Rightarrow\quad y &= -4 + 2(3) = 2 \\ x + y + z &= 6 &\Rightarrow\quad x &= 6 - 2 - 3 = 1 \end{aligned}$$

The solution is $(1, 2, 3)$, as before. This is the same calculation as the hand method in Section 6.8. Only the bookkeeping is different.

6.9.5. Reading off the special cases

The echelon form tells you which of the three cases you're in:

- A row of the form $[0 \quad 0 \quad 0 \mid k]$ with $k \neq 0$ says $0 = k$: **inconsistent**, no solution.
- A row of all zeros, $[0 \quad 0 \quad 0 \mid 0]$, says $0 = 0$: a redundant equation. If, after discarding such rows, there are fewer pivots than unknowns, the unknowns without a pivot are **free variables** and there are **infinitely many** solutions.
- A pivot in every unknown's column (and no contradiction): **exactly one** solution.

For a square system (as many equations as unknowns), that last case is the typical one, and in Volume 2, Chapter 27 you'll see it's the case where the coefficient matrix is **invertible**.

Example 37. Procedure — Gaussian elimination with back substitution

Input: an $n \times n$ system, written as the augmented matrix $[A \mid \mathbf{b}]$.

Steps:

1. For each column $j = 1, 2, \ldots, n$:
   a. **Pivot.** Among rows j to n, pick a row whose entry in column j is nonzero. By hand, prefer a ± 1 or a small integer. By computer, pick the one with the **largest absolute value** (*partial pivoting*). Swap it into row j. If every candidate is zero, the matrix is **singular**: stop and report "no unique solution".
   b. **Eliminate.** For each row i below j, compute the multiplier $m = a_{ij}/a_{jj}$ and do $R_i \leftarrow R_i - mR_j$. Column j is now zero below the pivot.
2. **Back-substitute.** For $i = n, n-1, \ldots, 1$: $x_i = \left(b_i - \sum_{k>i} a_{ik}x_k\right)/a_{ii}$, using the already-found x_k.

Output: the solution vector $(x_1, \ldots, x_n)$, or a report that the system is singular. In that case, inspect the echelon form for a contradiction (no solution) or free variables (infinitely many).

The symbol $\sum_{k>i}$ in the back-substitution step is **sigma notation**. It means "add up the following term for every k greater than i (and up to n)". In code it's a `sum(…)` over a range, as you'll see below. Sequences and sums get full treatment in Chapter 14.

6.9.6. Implementation

```python
def gaussian_solve(A, b):
    """Solve A x = b for square A by Gaussian elimination with partial pivoting.

    A is a list of n rows (each a list of n numbers); b is a list of n numbers.
    Returns the solution as a list. Raises ValueError if A is singular.
    """
    n = len(A)
    # Build the augmented matrix [A | b] as a fresh copy (don't mutate inputs).
    M = [list(map(float, row)) + [float(rhs)] for row, rhs in zip(A, b)]

    # Forward elimination: make zeros below each pivot.
    for col in range(n):
        # Partial pivoting: choose the row with the largest |entry| in this column.
        pivot_row = max(range(col, n), key=lambda r: abs(M[r][col]))
        if abs(M[pivot_row][col]) < 1e-12:
            raise ValueError("matrix is singular (no unique solution)")
        M[col], M[pivot_row] = M[pivot_row], M[col]
        for r in range(col + 1, n):
            factor = M[r][col] / M[col][col]
            for k in range(col, n + 1):
                M[r][k] -= factor * M[col][k]

    # Back substitution: solve from the last row upwards.
    x = [0.0] * n
    for i in range(n - 1, -1, -1):
        known = sum(M[i][k] * x[k] for k in range(i + 1, n))
        x[i] = (M[i][n] - known) / M[i][i]
    return x


A = [[1, 1, 1],
     [2, -1, 1],
     [1, 2, -1]]
b = [6, 3, 2]
print(gaussian_solve(A, b))
```

```
[1.0, 2.0000000000000004, 3.0000000000000004]
```

Notice the trailing `…0004`. Because the program pivots on the largest entry (2 in column 1, not 1), it divides by numbers like 2 and 3 along the way, and binary floating point can't represent $\frac{1}{3}$ exactly (Chapter 1). The answer is correct to about 16 significant digits, which is as good as `float` gets. Exact answers would need `fractions.Fraction` instead of `float`.



Why pivot on the largest entry?

Two failure modes motivate pivoting.

- **A zero pivot.** The system $0x + y = 2,\ x + y = 3$ has the unique solution $(1, 2)$, but naive elimination that always uses row 1 as the pivot divides by zero at once. Swapping rows fixes it.
- **A tiny pivot.** If the pivot is 10^{-20} instead of 0, there's no crash, but the multiplier is 10^{20} and rounding errors get amplified enormously. The program will return a confidently wrong answer.

Choosing the largest available pivot keeps every multiplier at most 1 in size and controls this growth. Real libraries always pivot. You'll see why in detail in Volume 2, Chapter 32 .

Computer-science connection: the cost of elimination

Count the work. Clearing column j touches about $(n - j)$ rows, each with about $(n - j)$ entries, so the forward phase costs roughly

$$\sum_{j=1}^{n} (n - j)^2 \approx \frac{n^3}{3}$$

multiply–add operations. Back substitution is only about $n^2/2$. So Gaussian elimination is $O(n^3)$: doubling the number of unknowns multiplies the work by eight. For $n = 1000$ that's about 3×10^8 operations, a fraction of a second on a laptop. For $n = 10^6$ it would be 3×10^{17}, which is why huge systems that are mostly zeros (**sparse** systems: circuits, finite-element meshes, PageRank) use specialised sparse or iterative solvers instead.

Two further facts worth knowing. First, the forward phase depends only on A, not on $\mathbf{b}$. Libraries record the multipliers as an **LU factorisation** and reuse it to solve for many right-hand sides at $O(n^2)$ each. Second, the algorithm can be done in place in the matrix's own memory, and its loops are the ones BLAS/LAPACK tune for cache behaviour. Every call to `numpy.linalg.solve` ends up in LAPACK's `gesv` routine, which is exactly this: LU factorisation with partial pivoting.

Gauss and the lost planet

On 1 January 1801 the astronomer Giuseppe Piazzi spotted a new object, the dwarf planet Ceres. He followed it for about 40 days before it disappeared into the Sun's glare. Astronomers expected it to be lost. The 24-year-old Carl Friedrich Gauss fitted an orbit to Piazzi's handful of observations, and in December 1801 Ceres was found almost exactly where he predicted. Fitting the orbit required solving systems of linear equations, many more equations than unknowns, in the least-squares sense (previewed in Section 6.11), and the elimination procedure Gauss described for it is the one now named after him.

Watch the staircase form

Display a 3×3 or 4×4 augmented matrix with a "step" button. Each press performs one row operation and highlights the pivot, the row being changed, and the multiplier. Alongside, a 3D view shows the three planes; after each row operation the planes **move** but their common point stays fixed. That's the geometric meaning of "equivalent system". Include a toggle between hand-friendly pivoting (smallest nonzero integer) and partial pivoting (largest absolute value), and a mode where the learner chooses each operation and the app checks it.

6.10. Exercises: three or more unknowns

1. **(Application)** Solve $x + y + z = 2$, $x - y + 2z = 9$, $2x + y - z = -3$.
2. **(Recognition)** Show that $x + y + z = 1$, $x + 2y + 3z = 2$, $2x + 3y + 4z = 4$ has no solution. What's the quickest way to see it?
3. **(Algorithm)** Explain what happens if you run Gaussian elimination **without** row swaps on $y = 2$, $x + y = 3$ (written as $0x + 1y = 2$ first). What does `gaussian_solve([[0, 1], [1, 1]], [2, 3])` return?
4. **(Problem solving)** Find the parabola $y = ax^2 + bx + c$ that passes through $(1, 2)$, $(2, 3)$ and $(3, 6)$.

▼ **Answers**

1. Number the equations (1), (2), (3) in order. Eliminate x. (2) − (1): $-2y + z = 7$. (3) − 2(1): $-y - 3z = -7$. Eliminate y from these two: twice the second is $-2y - 6z = -14$; subtract it from the first: $7z = 21$, so $z = 3$. Then $-y - 9 = -7$ gives $y = -2$, and $x = 2 - (-2) - 3 = 1$. Check: $1 + 2 + 6 = 9$ and $2 - 2 - 3 = -3$. Answer: $(1, -2, 3)$.
2. Add the first two equations: $2x + 3y + 4z = 3$. The third says $2x + 3y + 4z = 4$. The same left side can't equal both 3 and 4. In matrix terms, $R_3 \leftarrow R_3 - R_1 - R_2$ produces $[0 \quad 0 \quad 0 \mid 1]$, i.e. $0 = 1$. **Inconsistent.** (The three planes meet pairwise in parallel lines.)
3. Naive elimination would compute the multiplier $1/0$ and crash (or produce `inf` / `nan` with NumPy floats). With partial pivoting, the code swaps the rows first, solves $x + y = 3,\ y = 2$, and returns `[1.0, 2.0]`.

4. Substituting each point gives $a+b+c=2$, $4a+2b+c=3$, $9a+3b+c=6$. Subtract consecutive equations to eliminate c: $3a+b=1$ and $5a+b=3$. Subtract these: $2a=2$, so $a=1$, $b=1-3=-2$, and $c=2-1+2=3$. The parabola is $y=x^2-2x+3$. Check $(3,6)$: $9-6+3=6$. This is **polynomial interpolation**, which you'll meet again in Chapter 7.

> **US curriculum**
>
> Three-variable systems usually appear in **Algebra II** (grade 10 or 11). Writing systems as matrices and solving them by row reduction (often to **reduced** row echelon form, "RREF", on a graphing calculator) appears in Algebra II or **Precalculus**. Common Core lists matrix methods as "+" (advanced) standards **A-REI.C.8–9**. Gaussian elimination as an algorithm, with pivoting and cost analysis, is a first-year **university linear algebra** or **numerical methods** topic.

6.11. Applications, and solving with a library

6.11.1. Mixtures

A lab needs 40 litres of a 45% acid solution, and it stocks 30% and 70% solutions. How much of each should it mix?

Let x be the litres of 30% solution and y the litres of 70%. There are two facts, giving two equations:

$$\begin{aligned} x+y &= 40 && \text{(total volume)} \\ 0.30x+0.70y &= 0.45\times 40 = 18 && \text{(litres of pure acid)} \end{aligned}$$

Substitute $y=40-x$:

$$\begin{aligned} 0.30x+0.70(40-x) &= 18 \\ 0.30x+28-0.70x &= 18 \\ -0.40x &= -10 \\ x &= 25 \end{aligned}$$

So use 25 L of the 30% solution and 15 L of the 70% solution. Check: $7.5+10.5=18$ litres of acid. The same template, "one equation for the total amount, one for the total of some property", covers blending coffee beans, investing at two interest rates, and alloying metals.

6.11.2. Electrical circuits

Kirchhoff's voltage law says that around any closed loop in a circuit, the voltage supplied equals the sum of the voltage drops across the resistors, where a resistor R carrying current I drops RI volts (Ohm's law). Take a circuit with two loops. Loop 1 has a 10 V battery and a 2 Ω resistor. Loop 2 has a 6 Ω resistor. A 3 Ω resistor sits on the wire the two loops share. Assign a **loop current** I_1 to loop 1 and I_2 to loop 2. The shared resistor carries I_1-I_2:

$$\begin{aligned} \text{loop 1:}\quad 10 &= 2I_1+3(I_1-I_2) &\Rightarrow\quad 5I_1-3I_2 &= 10 \\ \text{loop 2:}\quad 0 &= 3(I_2-I_1)+6I_2 &\Rightarrow\quad -3I_1+9I_2 &= 0 \end{aligned}$$

The second equation gives $I_1 = 3I_2$. Substituting into the first: $15I_2 - 3I_2 = 10$, so $I_2 = \frac{10}{12} = \frac{5}{6}$ A, and $I_1 = \frac{5}{2} = 2.5$ A. Circuit simulators like SPICE do exactly this with thousands of loops or nodes, assembling one linear equation per loop or node and solving the resulting large sparse system.

6.11.3. Graphics: where do two lines cross?

In 2D graphics and games you constantly ask whether two moving things cross. Write a ray as a starting point plus a multiple of a direction. Ray 1 starts at $(0, 0)$ with direction $(4, 2)$, so its points are $(4t, 2t)$ for $t \geq 0$. Ray 2 starts at $(0, 3)$ with direction $(2, -1)$, so its points are $(2s, 3 - s)$. They meet where both coordinates agree:

$$\begin{cases} 4t = 2s \\ 2t = 3 - s \end{cases} \quad \Rightarrow \quad s = 2t, \quad 2t = 3 - 2t, \quad t = \tfrac{3}{4}, \quad s = \tfrac{3}{2}.$$

The crossing point is $(4 \cdot \frac{3}{4},\ 2 \cdot \frac{3}{4}) = (3, 1.5)$. The parameter values carry extra information. If ray 2 were a **segment** that ends at $s = 1$, then $s = \frac{3}{2}$ says the crossing lies beyond its end, so the segments **don't** hit. In 3D the same idea (ray meets triangle) is a 3×3 system, and it's the inner loop of ray tracing (Volume 2, Chapter 26).

6.11.4. Solving with NumPy

For real work you don't hand-code elimination. You call a library:

```python
import numpy as np

A = np.array([[1, 1, 1],
              [2, -1, 1],
              [1, 2, -1]], dtype=float)
b = np.array([6, 3, 2], dtype=float)
x = np.linalg.solve(A, b)
print(x)
print(np.allclose(A @ x, b))

# The two-loop circuit (mesh currents I1, I2 in amps)
C = np.array([[5, -3],
              [-3, 9]], dtype=float)
v = np.array([10, 0], dtype=float)
print(np.linalg.solve(C, v))
```

```
[1. 2. 3.]
True
[2.5        0.83333333]
```

Here `A @ x` is matrix–vector multiplication (Volume 2, Chapter 27). It recomputes every left-hand side, so `np.allclose(A @ x, b)` is the computer's version of "check in all the original equations".

So why learn it by hand? Four reasons:

1. **Setting up** the system is the hard part, and no library does it for you. Choosing unknowns and translating each fact into an equation is the real skill, and you can only check it if you understand what a solution means.
2. **Interpreting failure.** `np.linalg.solve` raises `LinAlgError: Singular matrix` for the parallel and coincident cases. Without knowing what singular means geometrically, you can't tell whether your model is missing an equation or contains two contradictory ones.
3. **Recognising structure.** Triangular, symmetric, banded, sparse: knowing how elimination works tells you when a specialised (and often thousands of times faster) solver applies.
4. **Judging trustworthiness**, which is the next topic.

6.11.5. A warning about nearly parallel lines

Consider *Try it:* Nudge b and watch x ; Nudge the nearly parallel lines .

$$\begin{cases} x + y = 2 \\ x + 1.001y = 2.001 \end{cases} \qquad \text{versus} \qquad \begin{cases} x + y = 2 \\ x + 1.001y = 2.002 \end{cases}$$

The only difference is one right-hand side, changed by 0.001. Subtracting the first equation from the second gives $0.001y = 0.001$ in the first system and $0.001y = 0.002$ in the second. So the first system has solution $(1, 1)$ and the second has solution $(0, 2)$. A change of one part in two thousand in the data moved the answer by a full unit in each coordinate.

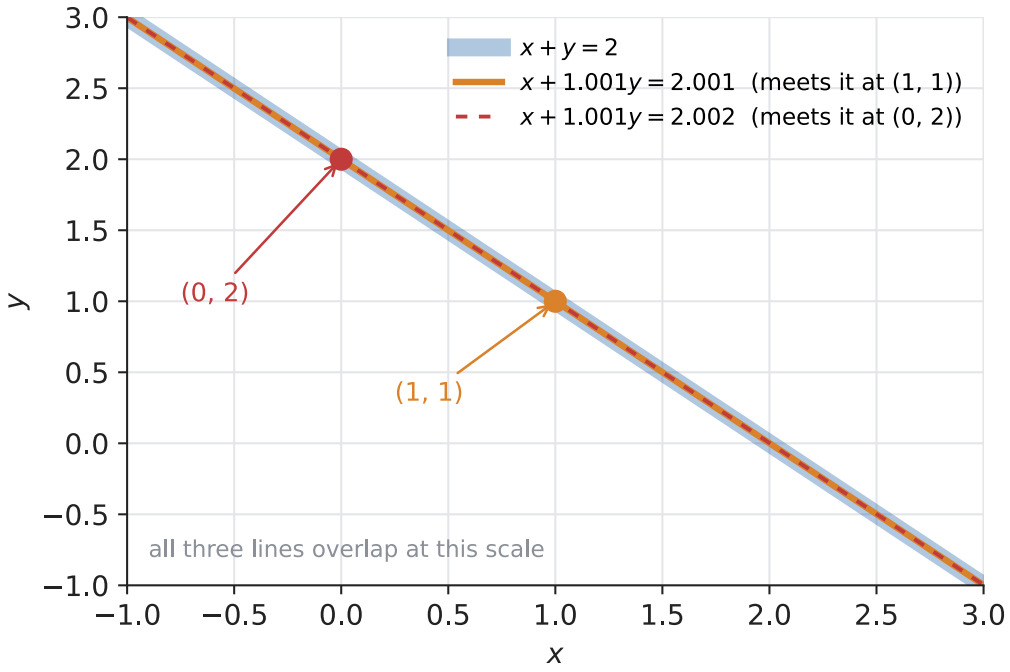


Figure 42. Two nearly parallel lines. Nudging one line by 0.001 slides the intersection from (1, 1) to (0, 2).

```python
import numpy as np

A = np.array([[1, 1],
              [1, 1.001]])
print(np.linalg.solve(A, [2, 2.001]))
print(np.linalg.solve(A, [2, 2.002]))   # right-hand side nudged by 0.001
print(round(np.linalg.cond(A)))
```

```
[1. 1.]
[0. 2.]
4002
```

The **condition number** (here about 4000) measures how much relative errors in the input can be amplified in the output. The library did nothing wrong: the **problem itself** is sensitive. If your inputs are measurements accurate to three digits, the answer to this system is essentially meaningless. Such a system is called **ill-conditioned**, and Volume 2, Chapter 32 explains how to measure and manage it. The geometric picture to keep in mind is two lines crossing at a shallow angle: the crossing point is poorly defined.

6.11.6. Regression preview: more equations than unknowns

Suppose you want the line $y = mx + b$ through the three data points $(0, 1)$, $(1, 3)$, $(2, 4)$. Each point gives one equation in the unknowns m, b:

$$\begin{cases} b = 1 \\ m + b = 3 \\ 2m + b = 4 \end{cases}$$

The first two give $b = 1, m = 2$, but then the third equation says $4 + 1 = 4$, which is false. Three equations in two unknowns is **overdetermined**, and real data almost never fits exactly. Instead you ask for the line that makes the total squared vertical error as small as possible: **least squares**. Minimising that error leads, as you'll derive in Volume 3, Chapter 31 , to a **square** 2×2 system called the **normal equations**:

$$\begin{cases} 3b + 3m = 8 \\ 3b + 5m = 11 \end{cases} \quad \Rightarrow \quad 2m = 3, \quad m = 1.5, \quad b = \tfrac{8 - 4.5}{3} = \tfrac{7}{6}.$$

```python
import numpy as np

xs = np.array([0, 1, 2], dtype=float)
ys = np.array([1, 3, 4], dtype=float)
A = np.column_stack([xs, np.ones_like(xs)])   # columns: x, 1  ->  unknowns m, b
(m, b), *_ = np.linalg.lstsq(A, ys, rcond=None)
print(round(m, 4), round(b, 4))
```

```
1.5 1.1667
```

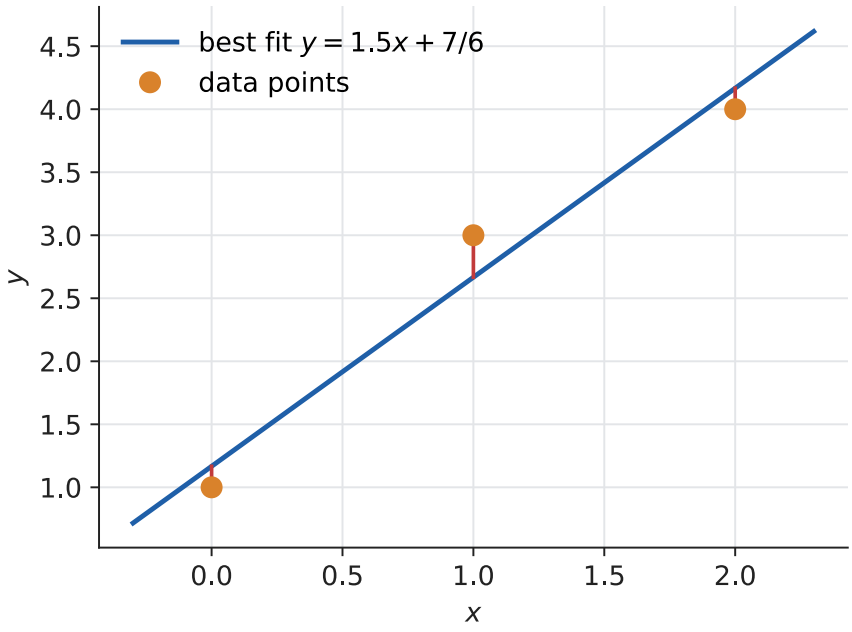


Figure 43. No line passes through all three points; least squares picks the line that minimises the sum of squared vertical gaps (red).

So "fit a model to data", the foundation of statistics and machine learning, is at heart "solve a linear system", once the overdetermined system has been turned into a square one.

Computer-science connection: linear systems everywhere

- **Graphics:** intersections (ray–plane, ray–triangle), solving for barycentric coordinates, and colour-space conversions are small linear systems solved millions of times per frame.
- **Simulation:** physics engines solve linear systems for contact forces every frame. Implicit time-stepping methods for cloth and fluids solve huge sparse ones.
- **Machine learning:** linear regression is a linear solve, and ridge regression adds one term to the diagonal. Newton-type optimisers solve a linear system at every step.
- **Search and ranking:** PageRank is the solution of a very large linear system (equivalently, an eigenvector problem; see Volume 2, Chapter 28).
- **Networks:** flows on networks (traffic, circuits, pipes) give one equation per node: "flow in = flow out".

In nearly all of these, the right question is not "can I solve it?" (the library can) but "is my system well-posed, well-conditioned, and structured enough to solve fast?"

US curriculum

Mixture, interest, and rate word problems with two unknowns are standard in **Algebra I**. Circuits appear in **physics** courses (AP Physics 1 and 2), usually solved by substitution. Linear regression on a calculator appears in Algebra I and **AP Statistics**, but the linear system behind it is normally left unexplained until **college linear algebra**.

6.12. Systems of linear inequalities and feasible regions

6.12.1. The problem

A small workshop makes tables and chairs. Each table earns \$30 profit and each chair \$20. The storeroom holds at most 40 items per week. Making a table takes 2 hours of labour and a chair 1 hour, and only 60 labour-hours are available. How many of each should it make to earn the most?

Let x be the number of tables and y the number of chairs. The constraints aren't equations, because you don't have to use all the storage or all the labour. They're **inequalities**:

$$\begin{cases} x + y \le 40 & \text{(storage)} \\ 2x + y \le 60 & \text{(labour)} \\ x \ge 0,\ y \ge 0 & \text{(can't make negative furniture)} \end{cases}$$

The goal is to make $P = 30x + 20y$ as large as possible.

6.12.2. Intuition: half-planes

The equation $x + y = 40$ is a line. The **inequality** $x + y \leq 40$ is that line together with **everything on one side of it**, which is called a **half-plane**. Which side? Test a point that's not on the line. The origin is usually easiest: $0 + 0 \leq 40$ is true, so the half-plane containing the origin is the one you want. (If the line passes through the origin, test a point like $(1, 0)$ instead.)

A **system** of inequalities asks for points satisfying **all** of them at once, which is the **intersection** of the half-planes. Graphically, shade each allowed region and look for where all the shadings overlap.



Half-plane, feasible region, vertex

- A **half-plane** is the set of points on one side of a line. It's **closed** (includes the boundary line, drawn solid) for $\leq$ or $\geq$, and **open** (boundary excluded, drawn dashed) for $<$ or $>$.
- The **feasible region** of a system of inequalities is the set of points satisfying every inequality. Points in it are **feasible**. If it's empty, the system is **infeasible**.
- A **vertex** (corner point) of the feasible region is a point where two boundary lines meet **and** every other constraint is satisfied. Finding each vertex means solving a 2×2 system of equations, which is where the earlier part of this chapter comes back in.

Example 38. Procedure — Graphing a system of linear inequalities

Input: linear inequalities in x and y.

Steps:

1. For each inequality, draw its boundary line (replace the inequality sign by $=$). Draw it solid for $\leq, \geq$ and dashed for $<, >$.
2. Test a point not on the line (usually $(0, 0)$). If it satisfies the inequality, that side is allowed; otherwise the other side is.
3. The feasible region is the overlap of all allowed sides.
4. Find each corner by solving the 2×2 system formed by the two boundary lines that meet there, then confirm the point satisfies every **other** inequality.

Output: the feasible region and its vertices (or the conclusion that it's empty).

6.12.3. Worked example: the workshop

Boundaries. $x + y = 40$ passes through $(40, 0)$ and $(0, 40)$. $2x + y = 60$ passes through $(30, 0)$ and $(0, 60)$. Testing $(0, 0)$ in both inequalities gives $0 \leq 40$ and $0 \leq 60$, both true, so we want the side towards the origin each time. Together with $x, y \geq 0$, the feasible region is a four-sided polygon in the first quadrant. *Try it:* See the workshop region .

Vertices.

- $(0, 0)$: where $x = 0$ meets $y = 0$.
- $(30, 0)$: where $y = 0$ meets $2x + y = 60$. Check storage: $30 \leq 40$. Feasible. (The other candidate on the x-axis, $(40, 0)$, fails labour: $80 > 60$.)
- $(0, 40)$: where $x = 0$ meets $x + y = 40$. Check labour: $40 \leq 60$. Feasible. (The candidate $(0, 60)$ fails storage.)
- The crossing of the two resource lines. Subtract $x + y = 40$ from $2x + y = 60$ to get $x = 20$, so $y = 20$. That's $(20, 20)$.

6.12.4. A taste of linear programming

Now the optimisation. For any target profit P, the points with $30x + 20y = P$ form a line with slope $-\frac{3}{2}$. Different values of P give **parallel** lines: **level lines** of the profit. Raising P slides the line outward, away from the origin. The largest profit you can actually achieve corresponds to the last level line that still touches the feasible region as it slides outward, and because the region is a polygon with straight edges, that last contact happens at a **corner**.

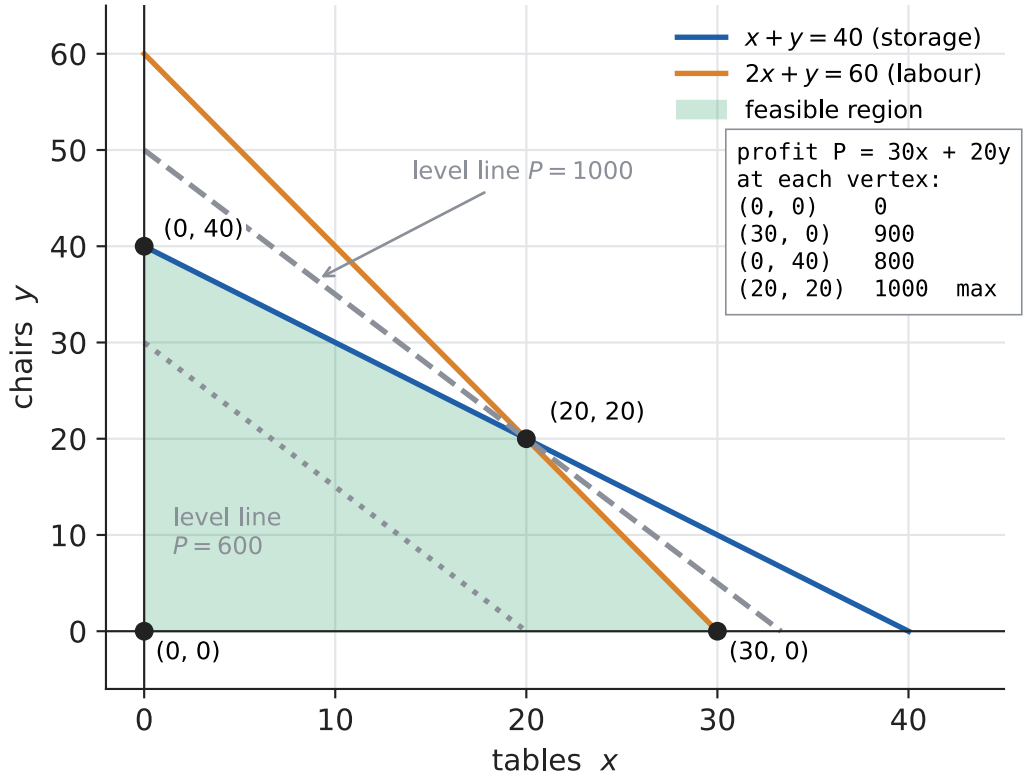


Figure 44. The workshop's feasible region (green), its four vertices with their profits, and two profit level lines. Pushing the level line outward, the last feasible point it touches is the vertex (20, 20).

So you only need to check the vertices:

Vertex (x, y)	Profit $P = 30x + 20y$
$(0, 0)$	0
$(30, 0)$	900
$(20, 20)$	$600 + 400 = 1000$
$(0, 40)$	800

The best plan is 20 tables and 20 chairs, for $1,000 profit. It uses all the storage ($20 + 20 = 40$) and all the labour ($40 + 20 = 60$). Both constraints are **binding**.



Corner-point principle

A **linear program** (LP) maximises or minimises a linear **objective function** (like $30x + 20y$) subject to linear inequality **constraints**. The **corner-point principle**: if an LP has an optimal solution and its feasible region has at least one vertex, then some vertex is optimal. (If the objective's level lines are parallel to an edge, every point on that edge ties.)

Here's the "check every vertex" method as code. It generates every pair of boundary lines, solves each 2×2 system, keeps the feasible points, and picks the best:

PYTHON

```
from itertools import combinations

# Each constraint is (a, b, c) meaning a*x + b*y <= c.
constraints = [(1, 1, 40),    # storage
               (2, 1, 60),    # labour
               (-1, 0, 0),    # x >= 0  written as  -x <= 0
               (0, -1, 0)]    # y >= 0
profit = lambda x, y: 30 * x + 20 * y

best = None
for (a1, b1, c1), (a2, b2, c2) in combinations(constraints, 2):
    det = a1 * b2 - b1 * a2
    if det == 0:
        continue                          # parallel boundaries never meet
    x = (c1 * b2 - b1 * c2) / det + 0.0   # "+ 0.0" turns -0.0 into 0.0
    y = (a1 * c2 - c1 * a2) / det + 0.0
    if all(a * x + b * y <= c + 1e-9 for a, b, c in constraints):
        print(f"vertex ({x:g}, {y:g})  profit {profit(x, y):g}")
        if best is None or profit(x, y) > profit(*best):
            best = (x, y)
print("best:", best, profit(*best))
```

```
vertex (20, 20)  profit 1000
vertex (0, 40)  profit 800
vertex (30, 0)  profit 900
vertex (0, 0)  profit 0
best: (20.0, 20.0) 1000.0
```

6.12.5. Failure modes and limitations



When linear programming goes wrong

- **Unbounded regions.** Drop the storage and labour limits and the profit can grow without limit. The LP has **no** optimum. Always check whether the feasible region is bounded in the direction you're optimising.
- **Empty regions.** Contradictory constraints ($x + y \le 10$ and $x + y \ge 20$) leave nothing feasible. That's the inequality version of an inconsistent system.
- **Integer requirements.** Our optimum $(20, 20)$ happened to be whole numbers. Often it isn't (e.g. $(13.7, 22.4)$ tables and chairs), and rounding can break a constraint or miss the true best whole-number plan. **Integer** linear programming is a much harder problem (NP-hard in general).
- **Brute force doesn't scale.** With m constraints in n variables there can be up to $\binom{m}{n}$ candidate vertices: hundreds of billions for modest real problems. The vertex-enumeration code above is for understanding only.

Computer-science connection: linear programming and constraint satisfaction

Real LP solvers use the **simplex method**, which walks from vertex to adjacent vertex, always improving the objective, and stops when no neighbour is better. Each step is a Gaussian-elimination-style **pivot**, the same word used for the same operation. Alternatively they use **interior-point methods**, which cut through the middle of the region. LPs schedule airline crews, route network traffic (max-flow is an LP; see Volume 3, Chapter 25), allocate cloud resources, and balance diets and portfolios. Libraries such as SciPy's `linprog`, Google OR-Tools, HiGHS and Gurobi do this at industrial scale.

More broadly, "find values satisfying all these constraints" is **constraint satisfaction**: type inference, package-version resolution (pip, npm, cargo), register allocation, and Sudoku are all examples. When the constraints are linear, the mathematics of this chapter gives you fast, exact tools. When they're not, you get SAT and SMT solvers (Volume 3, Chapter 21).

War, breakfast, and a Nobel Prize

Linear programming was developed independently by Leonid Kantorovich in the Soviet Union (1939, for planning plywood production) and George Dantzig in the US (1947, for planning US Air Force logistics). "Programming" here meant **planning** or **scheduling**, as in a "programme" of activities, not writing code. One of the first test problems was George Stigler's "diet problem": the cheapest combination of 77 foods meeting nine nutritional requirements. Kantorovich shared the 1975 Nobel Prize in economics for this work. Dantzig, famously, didn't.

Slide the profit line

The learner enters up to five linear inequalities. The app shades the feasible region and labels its vertices. A draggable slider for P moves the objective level line $30x + 20y = P$ across the plane, and the line turns red the moment it leaves the region. A second control rotates the objective (changing the profit ratio) so the learner can watch the optimal vertex jump from corner to corner, and see the tie when the objective lines up with an edge.

Try it: Slide the profit line

6.13. Exercises: inequalities and linear programming

1. **(Recognition)** Is the point $(3, 4)$ in the feasible region of $x + y \le 8$, $x - y \ge -2$, $y \ge 0$?
2. **(Application)** Maximise $P = 3x + 5y$ subject to $x + 2y \le 14$, $3x + 2y \le 18$, $x \ge 0$, $y \ge 0$. *Try it:* Check your answer .
3. **(Problem solving)** On the same region, **minimise** $C = x - y$. Then explain why maximising $P = x + y$ over the region $x \ge 0, y \ge 0, x - y \le 2$ has no answer.

▼ **Answers**

1. $3 + 4 = 7 \le 8$ ✓; $3 - 4 = -1 \ge -2$ ✓; $4 \ge 0$ ✓. Yes, it's feasible.
2. Vertices: $(0, 0)$; $(6, 0)$ (from $3x = 18$; check $6 \le 14$); $(0, 7)$ (from $2y = 14$; check $14 \le 18$); and the crossing of the two lines. Subtracting $x + 2y = 14$ from $3x + 2y = 18$ gives $2x = 4$, so $x = 2, y = 6$. Profits: $0, 18, 35, 6 + 30 = 36$. The maximum is 36 at $(2, 6)$.
3. At the vertices, $C = 0, 6, -7, -4$. The minimum is -7 at $(0, 7)$. In the second region, the points (t, t) for any $t \ge 0$ are feasible ($t - t = 0 \le 2$) with $P = 2t$, which grows without bound. The region is unbounded in the direction of increasing P, so no maximum exists.

US curriculum

Graphing linear inequalities in two variables and systems of them (Common Core **A-REI.D.12**) is part of **Algebra I**. Linear programming with the corner-point method appears in many **Algebra II** courses (and some Precalculus ones), usually as a word-problem unit. The simplex method and LP duality are **college** topics (operations research, algorithms courses).

6.14. Summary

- A **system** asks for values that satisfy every equation at once. Geometrically, for two unknowns, it's where lines cross. For three, it's where planes meet.
- There are always exactly three possibilities: one solution, no solution (the algebra produces $0 = \textbf{nonzero}$), or infinitely many (the algebra produces $0 = 0$; describe the solutions with a parameter).
- **Substitution** is best when a variable is already isolated or easy to isolate, and when an equation is nonlinear. **Elimination** is best for standard-form linear equations, and it's the version that scales.
- **Gaussian elimination** is elimination as an algorithm: three reversible row operations turn the augmented matrix into a staircase, followed by back substitution. It costs $O(n^3)$, it needs pivoting to be reliable, and it's what `numpy.linalg.solve` runs.
- Libraries solve; you **set up**, **interpret**, and **judge**. Nearly parallel lines mean an ill-conditioned problem, where small input errors produce large output errors.
- Systems of **inequalities** carve out feasible regions. A linear objective is optimised at a vertex, and vertices are 2×2 systems.

6.15. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Graphing calculator
- Linear systems, visually
- Where floating point bites

6.16. Mastery check

1. Solve by any method: $y = 2x + 1$, $3x - y = 4$. Say why you chose that method.
2. Solve by elimination: $4x - 3y = 1$, $6x + 5y = 49$.
3. Classify $6x - 9y = 12$, $4x - 6y = 8$, and describe the full solution set.
4. Solve $x + 2y - z = 3$, $2x - y + z = 5$, $x + y + 2z = 10$.
5. How many litres each of a 10% and a 25% saline solution make 30 litres of 20% solution?
6. Maximise $4x + 3y$ subject to $x + y \le 10$, $x \le 6$, $y \le 8$, $x, y \ge 0$.
7. A colleague's code calls `np.linalg.solve` on a 2×2 system and gets wildly different answers when the input data changes in the fourth decimal place. Give a geometric explanation, and say what single number you'd compute to confirm it.

Decision rule: All correct without hesitation → move on to Chapter 7 (or jump ahead to Volume 2, Chapter 27 if linear algebra is your priority). Stuck on 1–3 → reread Section 6.3, Section 6.4 and Section 6.5. Stuck on 4 → reread Section 6.8 and Section 6.9. Stuck on 5–7 → reread Section 6.11 and Section 6.12, then try the section exercises.

Try it: Check your answer .

▼ **Mastery check answers**

1. **Substitution**, because the first equation already gives y.
   $3x - (2x + 1) = 4 \Rightarrow x - 1 = 4 \Rightarrow x = 5$, then $y = 11$. Check: $15 - 11 = 4$. Answer: $(5, 11)$.
2. Multiply the first by 3 and the second by 2 to match the x-coefficients: $12x - 9y = 3$ and $12x + 10y = 98$. Subtract the first from the second: $19y = 95$, so $y = 5$. Then $4x - 15 = 1 \Rightarrow x = 4$. Check: $24 + 25 = 49$. Answer: $(4, 5)$.
3. Divide the first by 3 and the second by 2: both become $2x - 3y = 4$. Same line, so the system is **consistent and dependent**, with infinitely many solutions. With parameter t (so $y = t$): $(x, y) = \left(\frac{4+3t}{2},\ t\right), t \in \mathbb{R}$. For example, $t = 0$ gives $(2, 0)$.
4. Number the equations (1), (2), (3). Add (1) and (2) to eliminate z: $3x + y = 8$. Take 2 × (2) minus (3): $(4x - 2y + 2z) - (x + y + 2z) = 10 - 10$, i.e. $3x - 3y = 0$, so $x = y$. Then $4x = 8$, so $x = 2$ and $y = 2$. From (2): $z = 5 - 4 + 2 = 3$. Check (1): $2 + 4 - 3 = 3$; (3): $2 + 2 + 6 = 10$. Answer: $(2, 2, 3)$.
5. $x + y = 30$ and $0.10x + 0.25y = 0.20 \times 30 = 6$. Substitute $x = 30 - y$: $3 - 0.10y + 0.25y = 6 \Rightarrow 0.15y = 3 \Rightarrow y = 20$, so $x = 10$. That's 10 L of 10% and 20 L of 25%. Check: $1 + 5 = 6$.
6. Vertices: $(0, 0)$, $(6, 0)$, $(6, 4)$ (from $x = 6$ and $x + y = 10$), $(2, 8)$ (from $y = 8$ and $x + y = 10$), and $(0, 8)$. Objective values: $0, 24, 36, 32, 24$. The maximum is 36 at $(6, 4)$.

7. The two lines are nearly parallel, so they cross at a shallow angle, and a tiny shift in either line slides the crossing point a long way. The system is **ill-conditioned**. Compute the condition number, `np.linalg.cond(A)`. A large value (say 10^3 or more relative to the precision of the data) confirms it. The fix lies in the **model** (better data, a different formulation), not in a different solver. See Volume 2, Chapter 32 .

Chapter 7. Polynomials and Factoring

Chapter map: polynomials are the expressions you get when you are allowed to add, subtract and multiply, but never divide by the variable. They are the workhorse of algebra: they model areas, volumes, profits and trajectories; they are what computers evaluate when they approximate sine or exp; and they hide inside hash functions and network checksums. This chapter restores the vocabulary, the arithmetic, and the single most useful skill in elementary algebra, factoring, and treats each procedure as the algorithm it really is.

Where this chapter sits

You need: the rules of arithmetic and the distributive law from Chapter 1; fraction arithmetic from Chapter 2; exponent laws from Chapter 3; simplifying and expanding expressions from Chapter 4; and solving linear equations from Chapter 5.

This chapter unlocks: Chapter 8 (factoring is the first method for solving quadratics), Chapter 11 (polynomial functions are the first big family), Chapter 15 (every polynomial has a full set of complex roots), Volume 2, Chapter 17 (polynomials are the easiest functions to differentiate), Volume 3, Chapter 23 (polynomial arithmetic mirrors integer arithmetic), and Volume 2, Chapter 32 (Horner's method, interpolation).

7.1. What a polynomial is

7.1.1. The problem

You have a 20 cm by 30 cm sheet of card. You cut an x by x square out of each corner and fold up the flaps to make an open box. How big should x be to make the box hold as much as possible?

Write down the volume. The base is $(30 - 2x)$ long and $(20 - 2x)$ wide, and the box is x tall, so

$$V(x) = x(20 - 2x)(30 - 2x).$$

That is a perfectly good formula, but before you can do much with it (graph it, find where it is zero, find its maximum) you need to know what kind of object it is. Multiplied out, it becomes $V(x) = 4x^3 - 100x^2 + 600x$. That is a **polynomial**, and the whole of this chapter is about how to read, combine, take apart and evaluate expressions like it. (The maximum itself needs calculus; we return to this box in Volume 2, Chapter 18 .)

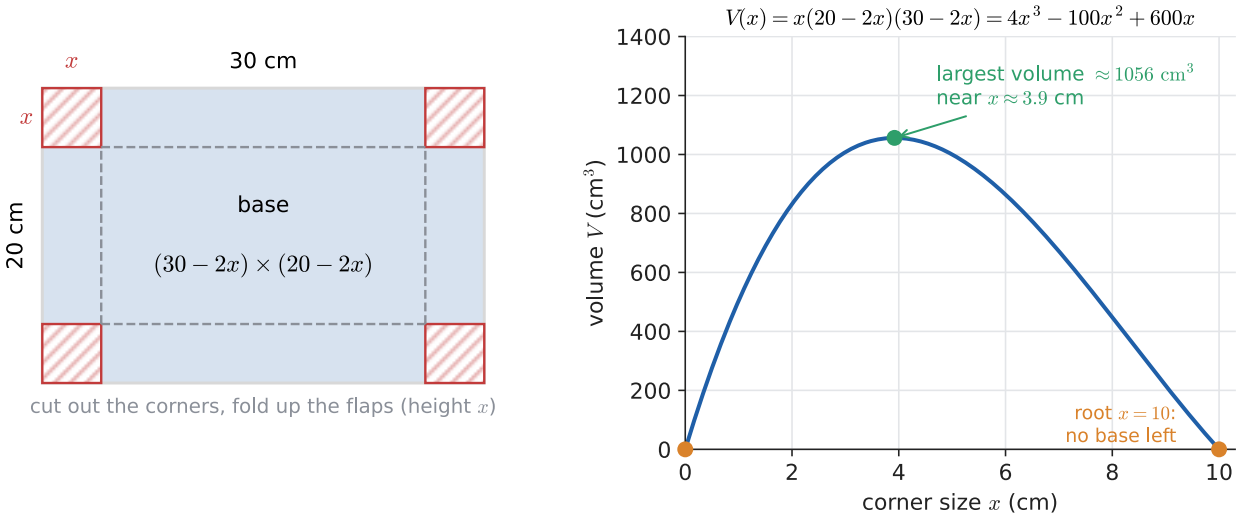


Figure 45. The open-box problem. Cutting x-by-x corners from a 30 cm by 20 cm card leaves a base $(30 - 2x)$ by $(20 - 2x)$; the volume is a cubic in x. Its roots $x = 0$ and $x = 10$ are the two "no box" extremes, and the largest volume lies between them.

7.1.2. Intuition

A polynomial is anything you can build from a variable and some numbers using only addition, subtraction and multiplication. Because multiplication by x repeated k times gives x^k, every polynomial can be rearranged into a sum of pieces of the form *number times a whole-number power of* x:

$$4x^3 - 100x^2 + 600x + 0.$$

Think of a polynomial as a list of numbers (the coefficients) attached to the slots $x^0, x^1, x^2, \ldots$. As a programmer you might store $4x^3 - 100x^2 + 600x$ as the array `[4, -100, 600, 0]`. That view, **a polynomial is its coefficient list**, turns out to be the key to almost every algorithm in this chapter.

What is *not* a polynomial? Anything with the variable in a denominator ($1/x$), under a root ($\sqrt{x}$), in an exponent (2^x), or inside a function like $\sin x$. Those have negative, fractional, or non-constant exponents, or no exponent form at all.

7.1.3. Vocabulary and notation

Term

A **term** is one of the pieces being added in an expression. In $7x^2 - 3x + 4$ the terms are $7x^2$, $-3x$ and 4. The sign belongs to the term: the middle term is $-3x$, not $3x$.

Coefficient

A **coefficient** is the number multiplying a power of the variable. In $7x^2 - 3x + 4$, 7 is the coefficient of x^2 and -3 is the coefficient of x. A missing power has coefficient 0; a bare x^2 has coefficient 1.

Constant term

The **constant term** is the term with no variable, i.e. the coefficient of x^0. In $7x^2 - 3x + 4$ it is 4. It is the value of the polynomial at $x = 0$.

Polynomial

A **polynomial** in x is an expression of the form

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

where n is a non-negative whole number and the a_k are constants (the coefficients). From Greek *poly* (many) plus Latin *nomen* (name, term): "many terms".

Degree

The **degree** of a polynomial is the highest power of the variable that appears with a non-zero coefficient. $4x^3 - 100x^2 + 600x$ has degree 3. A non-zero constant such as 5 has degree 0. Names: degree 1 **linear**, 2 **quadratic**, 3 **cubic**, 4 **quartic**, 5 **quintic**.

Leading term and leading coefficient

The **leading term** is the term of highest degree; its coefficient is the **leading coefficient**. In $-2x^5 + x^2 - 9$ the leading term is $-2x^5$ and the leading coefficient is -2. A polynomial whose leading coefficient is 1 is called **monic**.

Standard form

A polynomial is in **standard form** when like terms are combined and the terms are written in decreasing order of degree: $3x^4 - x^2 + 5x - 8$, not $5x - 8 + 3x^4 - x^2$.

Monomial, binomial, trinomial

Polynomials are also named by their number of terms (after combining like terms): a **monomial** has one term ($-6x^3$), a **binomial** two ($x^2 - 9$), a **trinomial** three ($x^2 + 5x + 6$). Beyond three we just say "polynomial".

Like terms

Like terms have the same variable raised to the same power: $3x^2$ and $-5x^2$ are like terms; $3x^2$ and $3x$ are not. Only like terms can be combined: $3x^2 - 5x^2 = -2x^2$.

Reading the notation. The general form $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ packs in a lot. The subscript on a_k is just a label, matching the power of x the coefficient sits in front of: a_3 is the coefficient of x^3. The dots $\cdots$ mean "carry on the pattern". So for $4x^3 - 100x^2 + 600x$ we have $n = 3$, $a_3 = 4$, $a_2 = -100$, $a_1 = 600$, $a_0 = 0$.

You will also see the same thing written with the summation sign Σ (capital Greek sigma, for "sum"):

$$p(x) = \sum_{k=0}^{n} a_k x^k.$$

Read this as: "let k run through the whole numbers from the value under the sigma (0) to the value on top (n); for each one write down $a_k x^k$; add them all up." It is exactly a `for k in range(0, n + 1)` loop accumulating a sum. The notation $p(x)$ names the polynomial and says what its variable is, so $p(2)$ means "substitute 2 for x".

7.1.4. Visualization: what the degree tells you about the graph

Far from the origin, the leading term dominates every other term: when $x = 1000$, $4x^3$ is four billion while $-100x^2$ is a mere hundred million. So the **ends** of a polynomial's graph are controlled by the leading term alone: its degree (even or odd) and the sign of its coefficient.

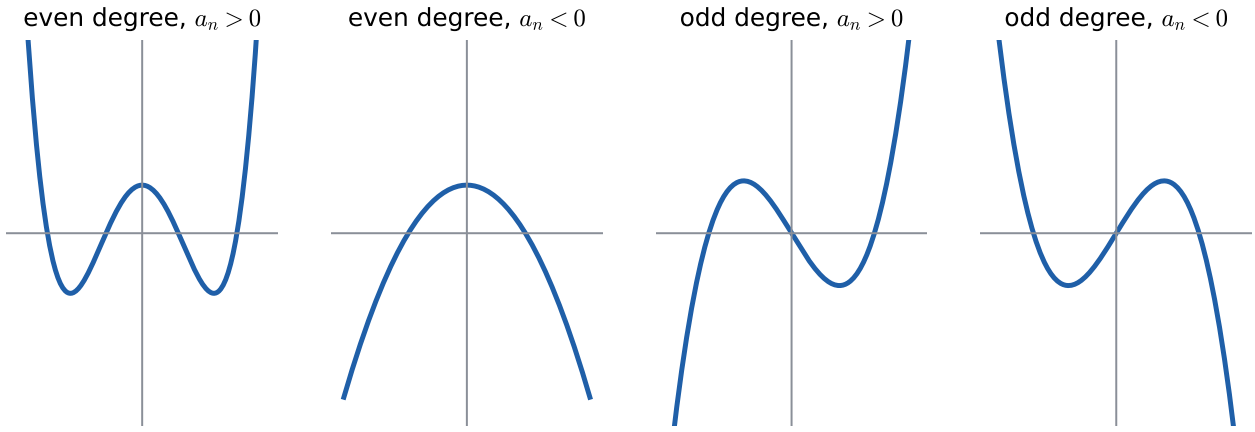


Figure 46. End behaviour is set by the leading term alone

Even degree: both ends go the same way (up if $a_n > 0$, down if $a_n < 0$). Odd degree: the ends go opposite ways. A polynomial of degree n can turn around at most $n - 1$ times, and crosses the x-axis at most n times; we will see why in Section 7.6.

7.1.5. Worked example: putting a polynomial in standard form

Write the box volume $V(x) = x(20 - 2x)(30 - 2x)$ in standard form, and state its degree, leading coefficient and constant term.

$$\begin{aligned}
(20 - 2x)(30 - 2x) &= 20 \cdot 30 + 20 \cdot (-2x) + (-2x) \cdot 30 + (-2x)(-2x) && \text{(distribute)} \\
&= 600 - 40x - 60x + 4x^2 \\
&= 4x^2 - 100x + 600 && \text{(combine like terms, order by degree)} \\
V(x) &= x\,(4x^2 - 100x + 600) \\
&= 4x^3 - 100x^2 + 600x && \text{(distribute the } x\text{)}
\end{aligned}$$

Degree 3 (a cubic), leading coefficient 4, constant term 0. The zero constant term tells you $V(0) = 0$: no cut, no box. That is a sanity check worth doing whenever you build a model.

7.1.6. Where you'd use this

Polynomials are the formulas you get from adding up products, so they turn up whenever quantities compound, accumulate or are approximated.

- **Saving.** Put \$1000 into an account at the start of each year for three years at interest rate r. Writing $g = 1 + r$ for the yearly growth factor, the balance at the end of year 3 is $1000g^3 + 1000g^2 + 1000g$, a cubic in g. At 5% ($g = 1.05$) that is $1157.625 + 1102.50 + 1050.00 = 3310.125$, or \$3310.13 to the nearest cent. Pension and loan calculators evaluate polynomials like this, and solving one for g (the "interest rate that makes the numbers work") is root-finding.
- **Mechanics and engineering.** Displacement under constant acceleration, $s = ut + \frac{1}{2}at^2$, is a polynomial in t. The sag of a cantilever beam under a uniform load is a quartic in the distance along the beam. Temperature sensors (thermocouples) are calibrated with published polynomials that turn a voltage into a temperature.
- **Computing.** Hardware has no "sine" instruction to lean on: math libraries reduce the angle to a small range and then evaluate a carefully chosen polynomial. Even three terms do well near 0: $x - \frac{x^3}{6} + \frac{x^5}{120}$ at $x = 0.5$ gives 0.4794271, against the true $\sin 0.5 = 0.4794255$. The curves in every font and vector drawing are cubic (or quadratic) polynomial pieces. Evaluating polynomials fast and accurately is the subject of Horner's method (Section 7.8).



> *Why "polynomial" and not "multinomial"?*
> "Polynomial" is a hybrid of Greek *poly* and Latin *nomen*, the kind of mongrel word purists hated. It was coined by analogy with "binomial", which goes back to the sixteenth century. Mathematicians before Descartes (1637) wrote polynomials in words or idiosyncratic abbreviations; Descartes's convention of letters near the end of the alphabet (x, y, z) for unknowns and letters near the start (a, b, c) for known quantities is the one we still use.

US curriculum

Polynomial vocabulary (term, degree, monomial/binomial/trinomial, standard form) and adding/subtracting/multiplying polynomials appear in **Algebra I** (grades 8–9). End behaviour and polynomial functions of higher degree are **Algebra II** topics (grades 10–11).

7.2. Adding, subtracting and multiplying polynomials

7.2.1. The problem

A company's revenue from selling q units is $R(q) = 40q - 0.01q^2$ and its cost is $C(q) = 5000 + 12q$. What is its profit? Profit is revenue minus cost, so you need to subtract one polynomial from another. Later, if the number of units sold is itself a polynomial in time, you will need to multiply and substitute. Polynomial arithmetic is how you combine models.

7.2.2. Intuition and why it works

Every rule here is a consequence of the ordinary laws of arithmetic you met in Chapter 1:

- **commutativity** ($a + b = b + a$, $ab = ba$) and **associativity** let you reorder and regroup terms;

- the **distributive law** $a(b+c) = ab + ac$ lets you combine like terms ($3x^2 + 5x^2 = (3+5)x^2$) and multiply out brackets;
- the **exponent law** $x^m \cdot x^n = x^{m+n}$ from Chapter 3 tells you what happens to powers when terms multiply.

Adding polynomials is adding coefficient lists slot by slot. **Subtracting** is the same after negating every coefficient of the second polynomial. **Multiplying** is "every term of the first times every term of the second, then collect".

7.2.3. Worked example: adding and subtracting

$$\begin{aligned}(3x^2 - 5x + 2) + (x^3 - 2x^2 + 7) &= x^3 + (3-2)x^2 + (-5)x + (2+7) && \text{(group like terms)}\\ &= x^3 + x^2 - 5x + 9.\end{aligned}$$

For subtraction, distribute the minus sign over the **whole** second bracket first:

$$\begin{aligned}(4x^3 - 2x + 1) - (x^3 + 3x^2 - 2x - 6) &= 4x^3 - 2x + 1 - x^3 - 3x^2 + 2x + 6 && \text{(negate every term)}\\ &= (4-1)x^3 - 3x^2 + (-2+2)x + (1+6) && \text{(group)}\\ &= 3x^3 - 3x^2 + 7.\end{aligned}$$

And the profit from the problem:
$P(q) = R(q) - C(q) = 40q - 0.01q^2 - 5000 - 12q = -0.01q^2 + 28q - 5000$, a quadratic.
(Chapter Chapter 8 finds its break-even points and its maximum.)



Distributing the minus sign

The single most common slip in polynomial arithmetic is writing $(4x^3 - 2x + 1) - (x^3 + 3x^2 - 2x - 6)$ as $4x^3 - 2x + 1 - x^3 + 3x^2 - 2x - 6$, negating only the first term. The minus sign multiplies the whole bracket, so **every** sign inside flips. If in doubt, rewrite the subtraction as adding $(-1) \times (\ldots)$.

7.2.4. Multiplication and the area model

To multiply two polynomials, multiply every term of the first by every term of the second and add the results. Why? Apply the distributive law twice:

$$\begin{aligned}(x+3)(x+2) &= x(x+2) + 3(x+2) && \text{(distribute over the first bracket)}\\ &= x^2 + 2x + 3x + 6 && \text{(distribute again)}\\ &= x^2 + 5x + 6. && \text{(combine like terms)}\end{aligned}$$

The picture makes this obvious. A rectangle $(x+3)$ wide and $(x+2)$ tall has area $(x+3)(x+2)$. Cut it along the joins and it falls into four pieces, one per pair of terms:

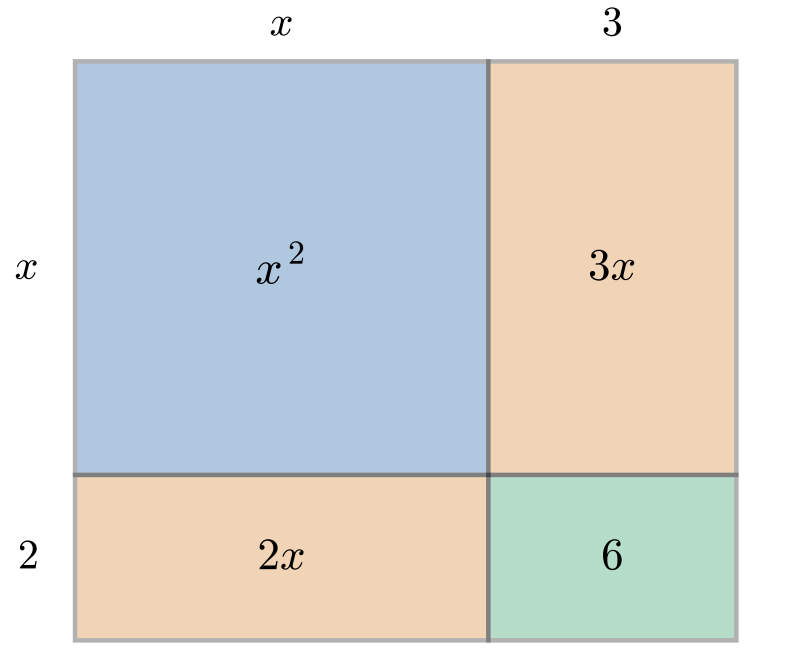


Figure 47. The area model: each cell is one term-by-term product

For longer polynomials the same idea becomes a **grid**: write the terms of one polynomial across the top, the other down the side, fill each cell with the product, then add along diagonals of equal degree. It is literally the long-multiplication algorithm from primary school, without the carries.



FOIL

FOIL (First, Outer, Inner, Last) is a US mnemonic for multiplying two binomials: $(a+b)(c+d) = ac + ad + bc + bd$. It is not a separate rule, just the distributive law (the four cells of a 2-by-2 area model) with a fixed order. It does **not** extend to longer polynomials; the grid does.

Example 39. Procedure — Multiplying polynomials

Input: polynomials $p(x)$ with terms $p_1, \dots, p_m$ and $q(x)$ with terms $q_1, \dots, q_k$.

Steps:

1. For every pair (p_i, q_j), multiply coefficients and add exponents: $ax^r \cdot bx^s = ab\,x^{r+s}$.
2. Collect the mk products.
3. Combine like terms and write the result in standard form.
4. Check: the degree of the product must be $\deg p + \deg q$, and the leading coefficient must be the product of the two leading coefficients.

Output: $p(x)\,q(x)$ in standard form.

7.2.5. Worked example: a binomial times a trinomial

Expand $(2x - 3)(x^2 + 4x - 5)$.

$$\begin{aligned}(2x-3)(x^2+4x-5) &= 2x\cdot x^2 + 2x\cdot 4x + 2x\cdot(-5) + (-3)\cdot x^2 + (-3)\cdot 4x + (-3)(-5) && \text{(six products)}\\ &= 2x^3 + 8x^2 - 10x - 3x^2 - 12x + 15 \\ &= 2x^3 + (8-3)x^2 + (-10-12)x + 15 && \text{(group like terms)}\\ &= 2x^3 + 5x^2 - 22x + 15.\end{aligned}$$

Check: degree $1 + 2 = 3$ ✓; leading coefficient $2 \cdot 1 = 2$ ✓; constant term $(-3)(-5) = 15$ ✓. A fourth check that catches almost everything: substitute an easy value. At $x = 1$ the left side is $(-1)(1 + 4 - 5) = 0$ and the right side is $2 + 5 - 22 + 15 = 0$ ✓.

7.2.6. Alternative approaches and method selection

- **Distribute term by term** (as above) is the default for anything up to about three terms by three.
- **Grid** is better for longer polynomials, because it is easy to see you have not missed a product.
- **Special-product patterns** (next section) are faster when they apply: $(x - 7)(x + 7)$ should be written down as $x^2 - 49$ without any work at all.
- **Coefficient-list convolution** (below) is what a computer does.

Computer-science connection: polynomial multiplication is convolution

Store a polynomial as its coefficient list, lowest power first or highest first, as long as you are consistent. Then the coefficient of x^k in the product $p \cdot q$ is

$$c_k = \sum_{i+j=k} a_i \, b_j,$$

"add up every product of a coefficient from p and one from q whose powers add to k". That operation on two lists has a name: **convolution**. It is the same operation as blurring an image with a kernel, computing a moving average, or combining two probability distributions of independent dice (Volume 3, Chapter 29).

PYTHON

```python
def poly_mul(a, b):
    """Multiply polynomials given as coefficient lists (highest power first)."""
    out = [0] * (len(a) + len(b) - 1)
    for i, ai in enumerate(a):
        for j, bj in enumerate(b):
            out[i + j] += ai * bj
    return out

print(poly_mul([2, -3], [1, 4, -5]))      # (2x - 3)(x^2 + 4x - 5)

import numpy as np
print(np.convolve([2, -3], [1, 4, -5]))   # NumPy calls it convolve
```

```
[2, 5, -22, 15]
[  2   5 -22  15]
```

The double loop costs $O(nm)$ multiplications. For very long polynomials (multiplying million-digit integers, which are polynomials in the base 10 or 2^{64}) the Fast Fourier Transform does it in $O(n \log n)$; see Volume 2, Chapter 32 .



Grid multiplier

Type two polynomials; the lab draws the area-model grid with each cell sized by its term (for positive values of x) and colours cells of equal degree the same. Slide x and watch the total area equal the product of the side lengths.

7.3. Special products

7.3.1. The problem

Some products come up so often that recognising them on sight saves enormous time, and, more importantly, lets you run them **backwards** when factoring. Can you compute 47×53 in your head? You can if you notice that $47 \times 53 = (50 - 3)(50 + 3)$.

7.3.2. The four patterns, and why they work

Perfect square. Expand by the distributive law:

$$(a+b)^2 = (a+b)(a+b) = a^2 + ab + ba + b^2 = a^2 + 2ab + b^2,$$
$$(a-b)^2 = a^2 - 2ab + b^2.$$

The picture: a square of side $a+b$ is made of an a^2 square, a b^2 square, and **two** $a \times b$ rectangles. The two rectangles are why the middle term is $2ab$.

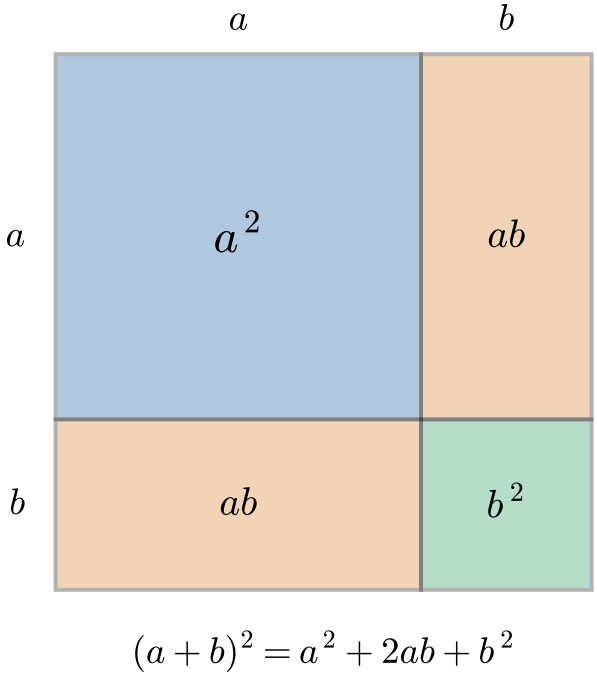


Figure 48. The perfect-square identity as an area

Difference of squares.

$$(a+b)(a-b) = a^2 - ab + ba - b^2 = a^2 - b^2.$$

The middle terms cancel. Geometrically: take an $a \times a$ square and remove a $b \times b$ corner. The L-shaped remainder can be cut into two rectangles and rearranged into a single rectangle $(a+b)$ long and $(a-b)$ high.

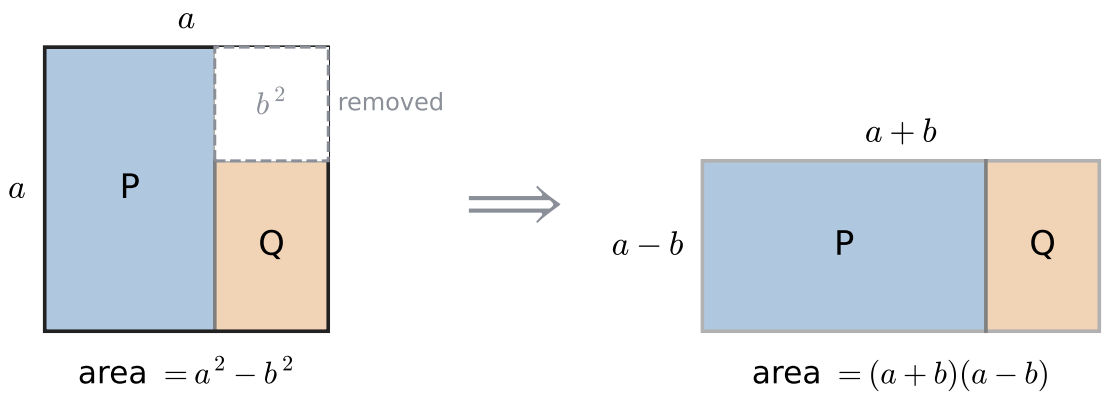


Figure 49. Difference of squares: cut the L-shape and re-assemble it

Sum and difference of cubes.

$$a^3 + b^3 = (a+b)(a^2 - ab + b^2),$$
$$a^3 - b^3 = (a-b)(a^2 + ab + b^2).$$

Verify the second by multiplying out, and watch four of the six terms cancel in pairs:

$$\begin{aligned}(a-b)(a^2+ab+b^2) &= a^3 + a^2b + ab^2 - a^2b - ab^2 - b^3 && \text{(six products)}\\ &= a^3 - b^3. && (\pm a^2b \text{ and } \pm ab^2 \text{ cancel})\end{aligned}$$

The first follows by replacing b with $-b$. A memory aid for the signs is "SOAP": the binomial has the **S**ame sign as the original, the next sign is the **O**pposite, and the last is **A**lways **P**ositive.



Special products

$$(a \pm b)^2 = a^2 \pm 2ab + b^2$$
$$(a+b)(a-b) = a^2 - b^2$$
$$a^3 \pm b^3 = (a \pm b)(a^2 \mp ab + b^2)$$

The $\pm$ ("plus or minus") and $\mp$ ("minus or plus") signs mean: read all top signs together, or all bottom signs together.

7.3.3. Worked examples

1. $47 \times 53 = (50-3)(50+3) = 50^2 - 3^2 = 2500 - 9 = 2491.$
2. $99^2 = (100-1)^2 = 10000 - 200 + 1 = 9801.$
3. $(3x-5)^2 = (3x)^2 - 2(3x)(5) + 5^2 = 9x^2 - 30x + 25.$
4. $(2x+7y)(2x-7y) = (2x)^2 - (7y)^2 = 4x^2 - 49y^2.$
5. $(x+2)(x^2 - 2x + 4) = x^3 + 2^3 = x^3 + 8.$



The freshman's dream

$(a+b)^2 \neq a^2 + b^2$. Test it: $(3+4)^2 = 49$ but $3^2 + 4^2 = 25$. The missing 24 is the two rectangles $2ab$ in the picture. Squaring does not distribute over addition, and neither does taking square roots: $\sqrt{9+16} = 5 \neq 3+4$. The same error in a compiler would be an invalid optimisation.



Mental arithmetic with algebra

Before electronic calculators, surveyors and astronomers used tables of quarter-squares to multiply: since $(a+b)^2 - (a-b)^2 = 4ab$,

$$ab = \frac{(a+b)^2}{4} - \frac{(a-b)^2}{4},$$

so one table of $n^2/4$ plus two lookups and a subtraction replaces a multiplication. Some early analogue and digital multipliers worked the same way.

US curriculum

Special products (perfect squares, difference of squares) are **Algebra I**; sum and difference of cubes usually wait for **Algebra II**.

7.4. Factoring

7.4.1. The problem

Solve $x^3 - 2x^2 - 5x + 6 = 0$.

There is no "isolate x" move here: x appears to three different powers. But suppose you knew that

$$x^3 - 2x^2 - 5x + 6 = (x+2)(x-1)(x-3).$$

Then the problem collapses. A product of numbers is zero exactly when one of the numbers is zero, so the solutions are $x = -2$, $x = 1$ and $x = 3$. Factoring turns one hard equation into several easy ones.



Zero-product property

If $AB = 0$ then $A = 0$ or $B = 0$ (or both). This is why factoring solves equations. It works **only** with zero on the right: $AB = 6$ tells you nothing about A or B individually.

7.4.2. Intuition

Factoring is multiplication run backwards. Expanding $(x+2)(x-1)(x-3)$ is mechanical; recovering the factors from $x^3 - 2x^2 - 5x + 6$ is a search problem, rather like factoring 391 into 17×23 is harder than multiplying. For polynomials there is no single formula that always works, but there is a reliable **decision procedure** that handles almost everything you meet by hand.

Factor

A **factor** of an expression is something that divides it exactly, so that the expression equals the factor times something else. $(x-3)$ is a factor of x^2-9 because $x^2-9=(x-3)(x+3)$. **To factor** (British: **to factorise**) means to rewrite a sum as a product.

Greatest common factor (GCF)

The **greatest common factor** of several terms is the largest expression dividing all of them. For $6x^3-9x^2$ it is $3x^2$, so $6x^3-9x^2=3x^2(2x-3)$. Take the GCF of the coefficients and the lowest power of each variable that appears in every term.

Irreducible (prime) polynomial

A polynomial is **irreducible** over a given number system if it cannot be written as a product of two polynomials of lower degree with coefficients in that system. x^2+4 is irreducible over the real numbers; x^2-2 is irreducible over the rationals but factors over the reals as $(x-\sqrt{2})(x+\sqrt{2})$. "Factor completely" means: keep going until every factor is irreducible (usually over the integers or rationals, unless stated).

7.4.3. The method: a decision procedure

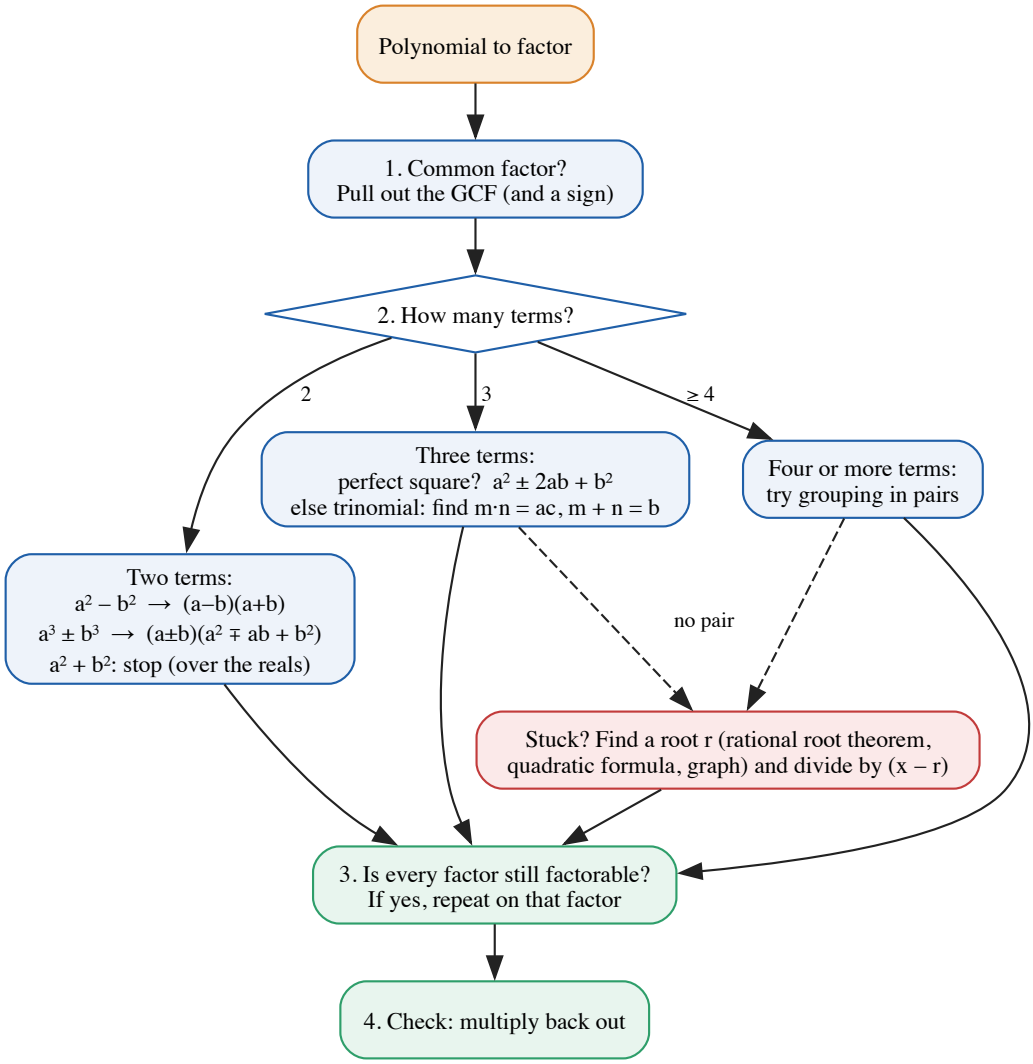


Figure 50. Factoring as a decision procedure

Example 40. Procedure — Factoring a polynomial completely

Input: a polynomial $p(x)$ with integer coefficients.

Steps:

1. **Common factor.** Pull out the GCF of all terms. If the leading coefficient is negative, pull out -1 too.
2. **Count the terms** of what remains.
   a. **Two terms:** look for $a^2 - b^2$, $a^3 - b^3$, $a^3 + b^3$. (A sum of squares $a^2 + b^2$ does not factor over the reals.)
   b. **Three terms:** look for a perfect square $a^2 \pm 2ab + b^2$; otherwise use the trinomial (ac) method below.
   c. **Four terms:** try grouping into pairs.

3. **Fallback:** if nothing fits, find a root r (rational root theorem, Section 7.7, or the quadratic formula for degree 2) and divide by $(x - r)$ (Section 7.5).
4. **Recurse:** apply the procedure again to every factor of degree 2 or more.
5. **Check** by multiplying back out (or substituting a test value).

Output: $p(x)$ as a product of irreducible factors.

7.4.4. The trinomial (ac) method

For a quadratic trinomial $ax^2 + bx + c$, you want to undo the area model: find the two middle cells. Their product must equal ac (the product of the two corner cells, ax^2 and c, has coefficient ac), and their sum must be b.

Example 41. Procedure — Factoring $ax^2 + bx + c$ (the ac method)

Input: integers a, b, c with $a \neq 0$.

Steps:

1. Compute ac.
2. Find integers m, n with $mn = ac$ and $m + n = b$. (List the factor pairs of ac; signs: if $ac > 0$ both have the sign of b; if $ac < 0$ they have opposite signs.) If no pair exists, the trinomial does not factor over the integers: stop.
3. Split the middle term: $ax^2 + mx + nx + c$.
4. Group in pairs and pull out the common factor of each pair; a common binomial appears.
5. Pull out the common binomial.

Output: $(px + q)(rx + s)$, or "irreducible over the integers".

When $a = 1$ the method simplifies: $x^2 + bx + c = (x + m)(x + n)$ where $mn = c$ and $m + n = b$. For $x^2 + 5x + 6$: $2 \times 3 = 6$, $2 + 3 = 5$, so $(x + 2)(x + 3)$.

7.4.5. Worked example: the ac method

Factor $6x^2 + 7x - 3$.

1. $ac = 6 \times (-3) = -18$. Negative, so m and n have opposite signs.
2. Factor pairs of -18 and their sums: $(-1, 18) \to 17$, $(1, -18) \to -17$, $(-2, 9) \to 7$ ✓. So $m = 9$, $n = -2$.
3. Split and group:

$$\begin{aligned} 6x^2 + 7x - 3 &= 6x^2 + 9x - 2x - 3 && \text{(split } 7x = 9x - 2x\text{)} \\ &= 3x(2x + 3) - 1(2x + 3) && \text{(factor each pair)} \\ &= (2x + 3)(3x - 1). && \text{(pull out } (2x + 3)\text{)} \end{aligned}$$

Check: $(2x + 3)(3x - 1) = 6x^2 - 2x + 9x - 3 = 6x^2 + 7x - 3$ ✓.

7.4.6. Worked examples: running the whole procedure

(a) $2x^3 - 8x$.

$$\begin{aligned} 2x^3 - 8x &= 2x(x^2 - 4) && \text{(GCF } 2x\text{)} \\ &= 2x(x - 2)(x + 2). && \text{(difference of squares)} \end{aligned}$$

(b) $x^4 - 16$.

$$\begin{aligned} x^4 - 16 = (x^2)^2 - 4^2 &= (x^2 - 4)(x^2 + 4) && \text{(difference of squares)} \\ &= (x - 2)(x + 2)(x^2 + 4). && \text{(again; } x^2 + 4 \text{ is a sum of squares: stop)} \end{aligned}$$

(c) $x^3 + 2x^2 - 9x - 18$ (four terms: group).

$$\begin{aligned} x^3 + 2x^2 - 9x - 18 &= x^2(x + 2) - 9(x + 2) && \text{(factor each pair)} \\ &= (x + 2)(x^2 - 9) && \text{(common binomial)} \\ &= (x + 2)(x - 3)(x + 3). && \text{(difference of squares)} \end{aligned}$$

(d) $3x^3 - 24$.

$$\begin{aligned} 3x^3 - 24 &= 3(x^3 - 8) && \text{(GCF)} \\ &= 3(x - 2)(x^2 + 2x + 4). && \text{(difference of cubes, } a = x,\ b = 2\text{)} \end{aligned}$$

The quadratic factor $x^2 + 2x + 4$ has $ac = 4$ with factor pairs summing to $4, 5, -4, -5$, never 2; it is irreducible over the integers (in fact over the reals, as Chapter 8 will show via its discriminant, $4 - 16 < 0$).

(e) $4x^2 - 12x + 9$. Three terms; first and last are squares ($(2x)^2$ and 3^2) and the middle is $-2(2x)(3)$. Perfect square: $(2x - 3)^2$.

7.4.7. Choosing a method

The order in the procedure is not arbitrary. Common factor first, because it shrinks everything else (try factoring $6x^2 + 42x + 72$ without first removing the 6). Patterns next, because they are instant when they fit. The ac method and grouping are systematic but slower. Root-finding plus division is the heavy machinery: it always works for a polynomial that has a rational root, but it is overkill for $x^2 - 9$.



Factoring traps

- **Stopping too early.** $x^4 - 16 = (x^2 - 4)(x^2 + 4)$ is not finished; $x^2 - 4$ still factors.
- **Forgetting the GCF**, then getting stuck in the ac method with huge numbers.
- **Cancelling a variable instead of factoring it out.** Dividing both sides of $x^3 = 4x$ by x loses the solution $x = 0$. Instead write $x^3 - 4x = x(x - 2)(x + 2) = 0$.
- **Most polynomials do not factor nicely.** $x^2 + x - 1$ has no integer factorisation. Textbook exercises are chosen to factor; real-world polynomials usually are not. That is why root formulas and numerical methods exist (Chapter 8, Volume 2, Chapter 32).

7.4.8. Where you'd use this

Factoring is how you solve "when does this polynomial equal zero?" by hand, via the zero-product property.

- **Dimensions from an area.** A rectangular vegetable bed must have area 96 m² and be 4 m longer than it is wide. With width x: $x(x + 4) = 96$, so $x^2 + 4x - 96 = 0$, which factors as $(x + 12)(x - 8) = 0$. The width is 8 m (the root -12 is meaningless here), so the bed is 8 m by 12 m.
- **Projectiles.** A ball kicked straight up at 20 m/s has height $h = 20t - 5t^2$ (taking $g \approx 10\ \text{m/s}^2$). Pulling out the common factor gives $h = 5t(4 - t)$, so it is at ground level at $t = 0$ and lands at $t = 4$ s. No formula needed: the factored form **is** the answer.
- **Computing.** The Reed–Solomon codes that let a scratched CD play and a smudged QR code scan detect where errors are by finding the roots of an "error-locator" polynomial (over a finite field), which amounts to factoring it into linear pieces.

Factoring is a search problem

Factoring integer polynomials by hand is a bounded search: the ac method enumerates factor pairs; the rational root theorem enumerates a finite candidate list. Computer-algebra systems use far cleverer algorithms (factoring modulo a prime, then "lifting"), and remarkably, factoring **polynomials** over the rationals can be done in polynomial time, while factoring large **integers** is believed to be hard, which is precisely what RSA encryption relies on (Volume 3, Chapter 23).



Factor explorer

Drag the two binomials $(px+q)$ and $(rx+s)$ by adjusting p, q, r, s; the lab shows the expanded trinomial and its area model. Then flip modes: given a trinomial, rearrange the tiles of an algebra-tile set into a rectangle, or discover that no rectangle exists.

7.5. Polynomial division

7.5.1. The problem

You know that $x = 1$ is a root of $x^3 - 2x^2 - 5x + 6$ (check: $1 - 2 - 5 + 6 = 0$). That should mean $(x - 1)$ is a factor. But what is the other factor? You need to **divide** $x^3 - 2x^2 - 5x + 6$ by $x - 1$. More generally, dividing polynomials is how you simplify rational expressions, find slant asymptotes (Chapter 11), and compute checksums.

7.5.2. Intuition: it is just long division

Recall long division of integers: to divide 7493 by 23, you ask "how many 23s fit into 74?", write 3, subtract 69, bring down the next digit, and repeat. The output is a **quotient** and a **remainder**, and

$$7493 = 23 \times 325 + 18, \qquad 0 \le 18 < 23.$$

Polynomial division is the same algorithm with "how many fit?" replaced by "what do I multiply the divisor's leading term by to match the current leading term?" It is actually **easier** than integer division, because there are no carries and no guessing.



Dividend, divisor, quotient, remainder

In $p(x) \div d(x)$, p is the **dividend** and d the **divisor**. Division produces a **quotient** $q(x)$ and **remainder** $r(x)$ such that

$$p(x) = d(x)\,q(x) + r(x), \qquad \deg r < \deg d \ (\text{or } r = 0).$$

If $r = 0$, d divides p exactly, and d is a factor of p.

7.5.3. The method: long division

Example 42. Procedure — Polynomial long division

Input: dividend $p(x)$ and non-zero divisor $d(x)$, both in standard form. **Insert a zero coefficient for every missing power in p.**

Steps:

1. Divide the leading term of the current dividend by the leading term of d. Write the result as the next term of the quotient.
2. Multiply the whole divisor by that term.
3. Subtract the product from the current dividend. (The leading term cancels, by construction.)
4. If the degree of what remains is still $\geq \deg d$, it becomes the new current dividend: go to step 1.

Output: quotient $q(x)$ (the terms you wrote down) and remainder $r(x)$ (what is left).

Each pass removes the leading term, so the degree drops by at least one each time; the loop must terminate after at most $\deg p - \deg d + 1$ passes.

7.5.4. Worked example: long division

Divide $2x^3 + 3x^2 - 4x + 5$ by $x^2 + 2x - 1$.

Pass 1. Leading terms: $2x^3 \div x^2 = 2x$. Multiply: $2x(x^2 + 2x - 1) = 2x^3 + 4x^2 - 2x$. Subtract:

$$(2x^3 + 3x^2 - 4x + 5) - (2x^3 + 4x^2 - 2x) = -x^2 - 2x + 5.$$

Pass 2. $-x^2 \div x^2 = -1$. Multiply: $-1 \cdot (x^2 + 2x - 1) = -x^2 - 2x + 1$. Subtract:

$$(-x^2 - 2x + 5) - (-x^2 - 2x + 1) = 4.$$

Degree 0 is less than 2, so stop. Quotient $2x - 1$, remainder 4:

$$2x^3 + 3x^2 - 4x + 5 = (x^2 + 2x - 1)(2x - 1) + 4.$$

Check: $(x^2 + 2x - 1)(2x - 1) = 2x^3 - x^2 + 4x^2 - 2x - 2x + 1 = 2x^3 + 3x^2 - 4x + 1$; add 4 to get the dividend ✓.

In the traditional layout this looks like integer long division:

```
                2x   -  1
             ____________________
x² + 2x - 1 ) 2x³ + 3x² - 4x + 5
              2x³ + 4x² - 2x
              ---------------
                  - x² - 2x + 5
                  - x² - 2x + 1
                  -------------
                              4
```

The algorithm, on coefficient lists:

PYTHON

```python
def poly_divide(num, den):
    """Long division of polynomials (coefficient lists, highest power first).

    Returns (quotient, remainder) as coefficient lists.
    """
    num = list(num)
    q = []
    while len(num) >= len(den):
        factor = num[0] / den[0]
        q.append(factor)
        for i in range(len(den)):
            num[i] -= factor * den[i]
        num.pop(0)                      # leading term is now zero
    return q, num

print(poly_divide([2, 3, -4, 5], [1, 2, -1]))
```

```
([2.0, -1.0], [0.0, 4.0])
```

The remainder list `[0.0, 4.0]` means $0x + 4$.

7.5.5. Synthetic division: the fast path for $(x - r)$

When the divisor is the linear factor $x - r$, long division simplifies dramatically. Every pass multiplies the divisor by a single number and subtracts; tracking only the coefficients, the whole thing collapses into one row of "multiply by r, add the next coefficient".

Example 43. Procedure — Synthetic division by $(x - r)$

Input: coefficients $a_n, a_{n-1}, \ldots, a_0$ of $p(x)$ (with zeros for missing powers) and the number r. Note: for divisor $x + 3$, $r = -3$.

Steps:

1. Bring down a_n: set $b_{n-1} = a_n$.
2. For each subsequent coefficient: multiply the last value by r and add the next coefficient.
3. The final value is the remainder; the earlier values are the quotient's coefficients (one degree lower than p).

Output: quotient coefficients and remainder.

7.5.6. Worked example: synthetic division

Divide $x^3 - 2x^2 - 5x + 6$ by $x - 1$ (so $r = 1$):

```
r = 1 |  1   -2   -5    6
      |       1   -1   -6
      +-------------------
         1   -1   -6  | 0
```

Step by step: bring down 1. $1 \times 1 + (-2) = -1$. $-1 \times 1 + (-5) = -6$. $-6 \times 1 + 6 = 0$. Remainder 0, quotient $x^2 - x - 6$. So

$$x^3 - 2x^2 - 5x + 6 = (x - 1)(x^2 - x - 6) = (x - 1)(x - 3)(x + 2),$$

finishing the problem we started with.

A second example with a missing term and a non-zero remainder: divide $2x^3 - 3x^2 + 5$ by $x - 2$. The coefficients are $2, -3, 0, 5$ (note the 0 for the missing x term).

```
r = 2 |  2   -3    0    5
      |       4    2    4
      +-------------------
         2    1    2  | 9
```

Quotient $2x^2 + x + 2$, remainder 9.

```python
def synthetic_division(coeffs, r):
    """Divide a polynomial by (x - r).

    coeffs: highest power first. Returns (quotient_coeffs, remainder).
    """
    out = [coeffs[0]]
    for c in coeffs[1:]:
        out.append(out[-1] * r + c)
    return out[:-1], out[-1]

print(synthetic_division([1, -2, -5, 6], 1))
print(synthetic_division([2, -3, 0, 5], 2))
```

```
([1, -1, -6], 0)
([2, 1, 2], 9)
```



Missing terms and the sign of r

- Leave a zero for every missing power. Dividing $2x^3 - 3x^2 + 5$ with coefficients `[2, -3, 5]` silently computes the wrong thing (it divides $2x^2 - 3x + 5$).
- Dividing by $x + 3$ means $r = -3$, not $+3$.
- Synthetic division as written only handles divisors $x - r$. For $2x - 1$, divide by $x - \frac{1}{2}$ and then divide the quotient by 2, or use long division.

Choosing a method. Linear divisor $x - r$: synthetic division, always. Anything else: long division.

US curriculum

Polynomial long division and synthetic division are **Algebra II** topics (grades 10–11), revisited in **Precalculus** for rational functions and asymptotes.

7.6. The remainder theorem, the factor theorem, and roots versus factors

7.6.1. The problem

In the last example, the remainder on dividing $2x^3 - 3x^2 + 5$ by $x - 2$ was 9. Now evaluate the polynomial at $x = 2$: $2(8) - 3(4) + 5 = 16 - 12 + 5 = 9$. Coincidence?

7.6.2. Why it works

Divide any polynomial $p(x)$ by $x - r$. The remainder has degree less than 1, so it is a constant, call it R:

$$p(x) = (x - r)\,q(x) + R.$$

This identity holds for every x. Substitute $x = r$:

$$p(r) = (r - r)\,q(r) + R = 0 \cdot q(r) + R = R.$$

That is the whole proof. Two consequences, each worth a name:



Remainder theorem

The remainder when $p(x)$ is divided by $(x - r)$ is $p(r)$.



Factor theorem

$(x - r)$ is a factor of $p(x)$ if and only if $p(r) = 0$. ("If and only if" means both directions hold: a factor gives a zero, and a zero gives a factor.)



Root and zero

A **root** of the equation $p(x) = 0$, also called a **zero** of the polynomial p, is a number r with $p(r) = 0$. On the graph $y = p(x)$, real roots are the x-coordinates where the curve meets the x-axis (the **x-intercepts**). "Root", "zero" and "solution" are used almost interchangeably; strictly, equations have **solutions** or **roots** and functions have **zeros**.

7.6.3. Roots and factors are the same information

The factor theorem says that **finding roots** and **finding linear factors** are the same task. This is the bridge between three representations of one object:

Representation	$p(x) = x^3 - 2x^2 - 5x + 6$
Symbols (standard form)	$x^3 - 2x^2 - 5x + 6$
Symbols (factored form)	$(x + 2)(x - 1)(x - 3)$
Words	"the cubic with roots -2, 1, 3 and leading coefficient 1"
Picture	crosses the x-axis at $-2, 1, 3$; meets the y-axis at $p(0) = 6$

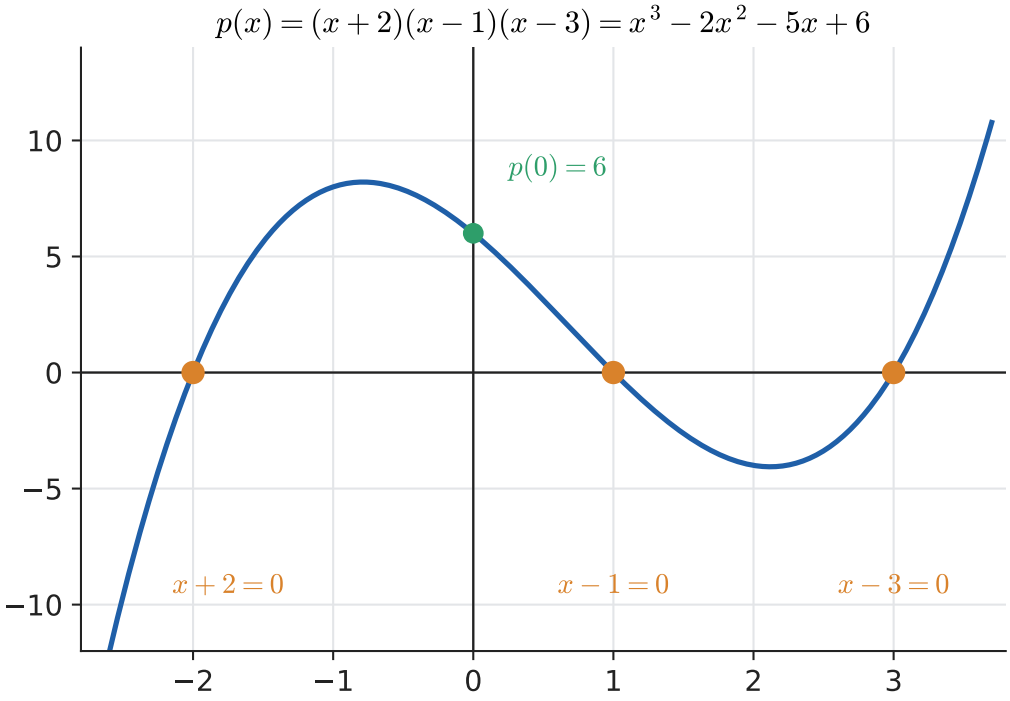


Figure 51. Factors, roots and intercepts are one fact seen three ways

Two important consequences:

- **A polynomial of degree n has at most n roots.** Each root r peels off a factor $(x - r)$ and drops the degree by one; after n peels you are left with a non-zero constant, which has no roots.
- **Multiplicity.** A factor can repeat. In $(x-1)^2(x+2)$ the root 1 has **multiplicity 2**. Graphically, at a root of even multiplicity the curve touches the axis and turns back; at odd multiplicity it crosses.



Every polynomial has all its roots, somewhere

$x^2 + 1$ has no real roots. But if you allow **complex** numbers (Chapter 15), the **fundamental theorem of algebra** (Gauss, 1799) guarantees that every polynomial of degree $n \geq 1$ has exactly n roots counted with multiplicity, so it factors completely into n linear factors. Over the reals, every polynomial factors into linear and irreducible quadratic pieces. And a theorem of Abel and Galois (1820s) shows that, unlike degrees 2, 3 and 4, there is **no** general formula in radicals for the roots of a degree-5 polynomial. Beyond degree 4, numerical methods are not a convenience but a necessity.

7.6.4. Worked example: using the factor theorem

Show that $(x+3)$ is a factor of $p(x) = x^3 + 4x^2 + x - 6$, then factor completely.

$(x+3) = (x-(-3))$, so test $r = -3$:

$$p(-3) = (-27) + 4(9) + (-3) - 6 = -27 + 36 - 3 - 6 = 0. \checkmark$$

Synthetic division by $r = -3$: bring down 1; $1(-3) + 4 = 1$; $1(-3) + 1 = -2$; $-2(-3) - 6 = 0$. Quotient $x^2 + x - 2 = (x+2)(x-1)$. Therefore $p(x) = (x+3)(x+2)(x-1)$.

7.6.5. Building a polynomial from its roots

The factor theorem also runs forwards. "Find a cubic with roots 1, 2, -4 and $p(0) = 16$" means $p(x) = k(x-1)(x-2)(x+4)$ for some constant k. At $x = 0$: $k(-1)(-2)(4) = 8k = 16$, so $k = 2$. This idea, choosing a polynomial to pass through prescribed points, grows into **polynomial interpolation** (Lagrange interpolation), the basis of Shamir's secret-sharing scheme and Reed–Solomon error-correcting codes (Volume 2, Chapter 32).

7.6.6. Where you'd use this

- **Engineering: is the system stable?** The behaviour of a feedback system (a cruise control, a drone's attitude controller, an amplifier) is governed by the roots of its **characteristic polynomial**. The system settles down when every root is negative (more precisely, has negative real part). Suppose testing reveals that $s = -1$ is a root of $s^3 + 6s^2 + 11s + 6$. Synthetic division by $(s+1)$ leaves $s^2 + 5s + 6 = (s+2)(s+3)$, so the roots are $-1, -2, -3$: all negative, so the system is stable. Dividing out a known root to shrink the problem is called **deflation**, and numerical root-finders do it routinely.
- **Signal processing: designing from the roots.** Building a polynomial from chosen roots is how simple digital filters are designed. To block a particular frequency (for example, 50 Hz or 60 Hz mains hum in an audio recording), the designer places a pair of (complex) roots of the filter's polynomial at the point corresponding to that frequency and multiplies out the factors to get the coefficients the code actually uses.
- **Checking data.** The remainder theorem says the remainder on division by $x - r$ is just $p(r)$. Checksums such as CRCs use the remainder on division by a fixed polynomial as a fingerprint of a message (Section 7.10).

7.7. The rational root theorem

7.7.1. The problem

The factor theorem is only useful if you can **find** a root. Trying random numbers is hopeless. For polynomials with integer coefficients, the rational root theorem gives a finite list of candidates.

7.7.2. Why it works

Suppose the fraction p/q, in lowest terms, is a root of

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0$$

with integer coefficients. Substitute and multiply through by q^n:

$$a_n p^n + a_{n-1} p^{n-1} q + \cdots + a_1 p q^{n-1} + a_0 q^n = 0.$$

Move the last term across: $a_0 q^n = -p\left(a_n p^{n-1} + \cdots + a_1 q^{n-1}\right)$. The right side is a multiple of p, so p divides $a_0 q^n$. But p shares no factor with q (lowest terms), so p must divide a_0. The same argument with the **first** term moved across shows q divides a_n.

Rational root theorem

If a polynomial with integer coefficients has a rational root p/q in lowest terms, then p divides the constant term a_0 and q divides the leading coefficient a_n. For a **monic** polynomial ($a_n = 1$), every rational root is an integer dividing a_0.

Example 44. Procedure — Finding rational roots

Input: a polynomial with integer coefficients and non-zero constant term. (If $a_0 = 0$, factor out x first.)

Steps:

1. List the divisors p of a_0 and the divisors q of a_n.
2. Form all candidates $\pm p/q$ and remove duplicates.
3. Test candidates (easiest first) by evaluating $p(x)$, ideally by synthetic division, which gives the quotient for free when the remainder is zero.
4. When a root is found, continue with the quotient (lower degree). Once the quotient is quadratic, factor it or use the quadratic formula.

Output: all rational roots, and a factorisation.

7.7.3. Worked example

Solve $2x^3 - 3x^2 - 11x + 6 = 0$.

Divisors of $a_0 = 6$: $1, 2, 3, 6$. Divisors of $a_n = 2$: $1, 2$. Candidates:

$$\pm 1,\ \pm 2,\ \pm 3,\ \pm 6,\ \pm \tfrac{1}{2},\ \pm \tfrac{3}{2}.$$

Test $x = 1$: $2 - 3 - 11 + 6 = -6$ ✗. $x = -1$: $-2 - 3 + 11 + 6 = 12$ ✗. $x = 2$: $16 - 12 - 22 + 6 = -12$ ✗. $x = 3$: $54 - 27 - 33 + 6 = 0$ ✓.

Synthetic division by 3: bring down 2; $2 \cdot 3 - 3 = 3$; $3 \cdot 3 - 11 = -2$; $-2 \cdot 3 + 6 = 0$. Quotient $2x^2 + 3x - 2$. Factor with $ac = -4$: $4 \cdot (-1) = -4$, $4 + (-1) = 3$:

$$2x^2 + 3x - 2 = 2x^2 + 4x - x - 2 = 2x(x + 2) - (x + 2) = (2x - 1)(x + 2).$$

So $2x^3 - 3x^2 - 11x + 6 = (x - 3)(2x - 1)(x + 2)$, and the roots are $x = 3, \tfrac{1}{2}, -2$.

PYTHON

```python
from fractions import Fraction

def divisors(n):
    n = abs(n)
    return [d for d in range(1, n + 1) if n % d == 0]

def rational_roots(coeffs):
    """All rational roots of a polynomial with integer coefficients.

    coeffs: highest power first; assumes the constant term is non-zero.
    """
    lead, const = coeffs[0], coeffs[-1]
    candidates = {Fraction(sign * p, q)
                  for p in divisors(const)
                  for q in divisors(lead)
                  for sign in (1, -1)}
    def value(x):
        result = 0
        for c in coeffs:
            result = result * x + c
        return result
    return sorted(r for r in candidates if value(r) == 0)

print(rational_roots([2, -3, -11, 6]))
```

```
[Fraction(-2, 1), Fraction(1, 2), Fraction(3, 1)]
```

`Fraction` does exact rational arithmetic, so "equals zero" is a reliable test here. With floating point it would not be (Volume 2, Chapter 32).



Limits of the rational root theorem

The theorem gives **candidates**, not roots; most fail. And it says nothing about irrational roots: $x^3 - 2 = 0$ has candidates $\pm 1, \pm 2$, none of which work, because its only real root is $\sqrt[3]{2}$. When every candidate fails, the polynomial has no rational roots, and you move on to the quadratic formula (degree 2) or numerical methods.

US curriculum

The remainder and factor theorems, the rational root theorem, and the link between zeros and factors are standard **Algebra II** content (the Common Core standard "A-APR.B: understand the relationship between zeros and factors of polynomials"). Multiplicity and end behaviour return in **Precalculus**.

7.8. Evaluating polynomials fast: Horner's method

7.8.1. The problem

Every time a computer evaluates $\sin x$, e^x or $\ln x$ in a math library, it is (after some range reduction) evaluating a carefully chosen polynomial. Graphics shaders evaluate polynomials for every pixel. So: what is the cheapest way to compute $p(x) = a_n x^n + \cdots + a_1 x + a_0$?

7.8.2. Intuition

The obvious way computes each power separately: x^n takes $n - 1$ multiplications, then one more for the coefficient. Summing over all terms, that is about $n(n+1)/2$ multiplications. Reusing powers (compute x^2 from x, x^3 from x^2) brings it down to about $2n$. But you can do better by **factoring out x repeatedly**:

$$\begin{aligned} 2x^3 - 6x^2 + 2x - 1 &= \big(2x^2 - 6x + 2\big)x - 1 \\ &= \big((2x - 6)x + 2\big)x - 1 \\ &= \Big(\big((2)x - 6\big)x + 2\Big)x - 1. \end{aligned}$$

Reading from the inside out: start with the leading coefficient; repeatedly **multiply by x and add the next coefficient**. That is exactly the synthetic-division row from Section 7.5, and its final value is the remainder on dividing by $x - r$, which by the remainder theorem is $p(r)$. Horner's method and synthetic division are the same algorithm.

Example 45. Procedure — Horner's method

> **Input:** coefficients $a_n, \ldots, a_0$ (highest first) and a value x.
>
> **Steps:**
>
> 1. Set $\mathbf{result} \leftarrow a_n$.
> 2. For $k = n - 1$ down to 0: $\mathbf{result} \leftarrow \mathbf{result} \times x + a_k$.
>
> **Output:** $\mathbf{result} = p(x)$. Cost: exactly n multiplications and n additions.

7.8.3. Worked example

Evaluate $p(x) = 2x^3 - 6x^2 + 2x - 1$ at $x = 3$.

Step	Computation	Result
start	a_3	2
1	$2 \times 3 + (-6)$	0
2	$0 \times 3 + 2$	2
3	$2 \times 3 + (-1)$	5

So $p(3) = 5$. Direct check: $2(27) - 6(9) + 2(3) - 1 = 54 - 54 + 6 - 1 = 5$ ✓. Three multiplications instead of the six needed naively.

PYTHON

```
def horner(coeffs, x):
    """Evaluate a polynomial at x.

    coeffs lists coefficients from the highest power down:
    [2, -6, 2, -1] means 2x^3 - 6x^2 + 2x - 1.
    """
    result = 0
    for c in coeffs:
        result = result * x + c
    return result

print(horner([2, -6, 2, -1], 3))
```

```
5
```

(Starting from `result = 0` makes the first pass give $0 \cdot x + a_n = a_n$, which keeps the loop uniform.)

7.8.4. Operation counts and why they matter

Method (degree n)	Multiplications	Additions
Each power from scratch	$n(n+1)/2$	n
Running power $x^k = x^{k-1} \cdot x$	$2n - 1$	n
Horner	n	n

For degree 10 that is 55 versus 19 versus 10 multiplications. Horner is provably optimal for general polynomials (Ostrowski 1954, Pan 1966): no algorithm can evaluate an arbitrary degree-n polynomial with fewer than n multiplications. Modern CPUs have a **fused multiply–add** (FMA) instruction computing $ax + b$ in one step with one rounding, and Horner's loop is nothing but a chain of FMAs. Horner is also usually **more accurate** in floating point, because it avoids computing huge powers that are then partially cancelled.

Computer-science connection: Horner is everywhere

- **Parsing numbers.** Converting the string `"4721"` to an integer is `n = n*10 + digit` in a loop: Horner's method evaluating the polynomial $4x^3 + 7x^2 + 2x + 1$ at $x = 10$. A number written in base b **is** a polynomial evaluated at b.
- **Hashing strings.** Java's `String.hashCode()` is $s_0 \cdot 31^{n-1} + s_1 \cdot 31^{n-2} + \cdots + s_{n-1}$ (mod 2^{32}), computed by Horner with $x = 31$. See Section 7.10.
- **Math libraries.** `exp`, `sin` and friends end in a Horner evaluation of a minimax polynomial (Volume 2, Chapter 32).



Horner versus the naive loop

Pick a degree n and a value of x. The lab evaluates the polynomial three ways, counts multiplications, and (in floating point, for polynomials like $(x-1)^{10}$ expanded out) plots the error of each method near $x = 1$, where cancellation makes the expanded form noisy whichever way you evaluate it: the naive method is worst and Horner is somewhat better, while evaluating the factored form $(x-1)^{10}$ directly stays accurate.

Try it: Horner versus the naive loop



Horner, and the people before him

William George Horner published the method in 1819, but it was known to Paolo Ruffini a few years earlier, to Isaac Newton in the 1660s, and to Chinese mathematicians (Qin Jiushao, 1247) centuries before that, who used it as part of a method for extracting roots numerically. It is a good example of an algorithm being named after someone who popularised it in the West.

7.9. Rational expressions

7.9.1. The problem

A cyclist rides 30 km to a town at speed v km/h and returns at $v + 5$ km/h. Total time:

$$T(v) = \frac{30}{v} + \frac{30}{v+5}.$$

To work with this (solve $T = 5$, say) you need to combine fractions whose numerators and denominators are polynomials. These are **rational expressions**, and they behave exactly like the numerical fractions of Chapter 2, with one extra hazard: the denominator can be zero for some values of the variable.

Rational expression

A **rational expression** is a quotient of two polynomials, $\dfrac{P(x)}{Q(x)}$ with Q not the zero polynomial. "Rational" here means "ratio", as in rational numbers p/q.

Domain

The **domain** of an expression is the set of values of the variable for which it is defined. For a rational expression, that is every real number **except** the roots of the denominator. The domain of $\dfrac{x+1}{x^2-4}$ is all real x except $x = \pm 2$, written $x \neq \pm 2$.

7.9.2. Simplifying: cancel factors, never terms

Just as $\frac{6}{8} = \frac{2\cdot 3}{2\cdot 4} = \frac{3}{4}$, you simplify a rational expression by factoring numerator and denominator and cancelling **common factors**:

$$\begin{aligned} \frac{x^2-4}{x^2-x-2} &= \frac{(x-2)(x+2)}{(x-2)(x+1)} && \text{(factor both)} \\ &= \frac{x+2}{x+1}, \qquad x \neq 2,\ x \neq -1. && \text{(cancel } (x-2)\text{)} \end{aligned}$$

The restriction $x \neq 2$ must be carried along: the original expression is undefined at $x = 2$ (it is $0/0$), even though the simplified one is not. The two expressions are equal **on the original domain**, and only there.

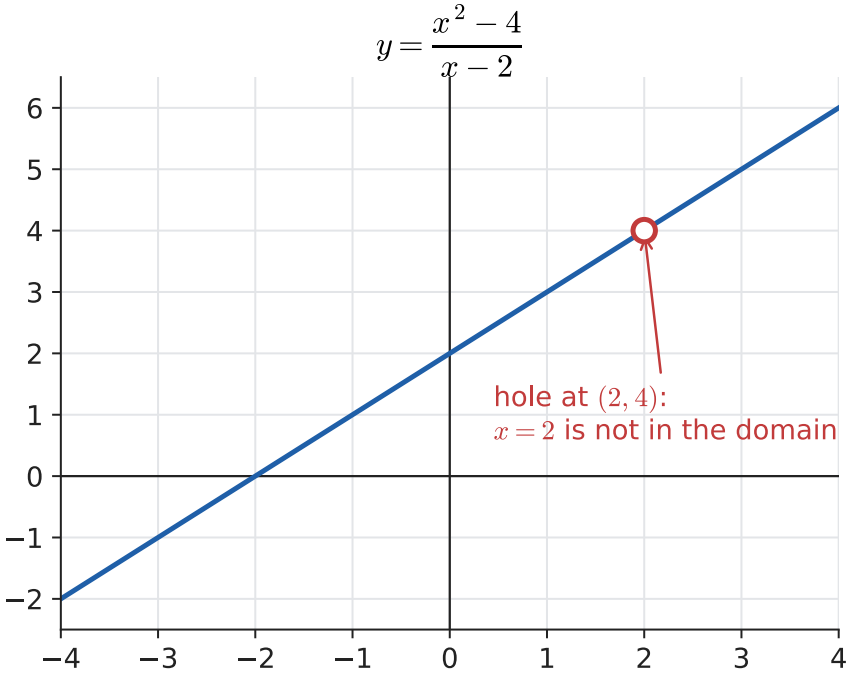


Figure 52. Cancelling a factor leaves a hole

The simplest case shows the picture. Graphically, $\frac{x^2-4}{x-2}$ is the line $y = x + 2$ with a single point punched out at $(2, 4)$: a **removable discontinuity**, or **hole**. The example above behaves the same way: $\frac{x+2}{x+1}$ with a hole at $(2, \frac{4}{3})$. A factor that does **not** cancel from the denominator produces a **vertical asymptote** instead; that story is told in Chapter 11 and Volume 2, Chapter 16 .

Example 46. Procedure — Simplifying a rational expression

Input: $P(x)/Q(x)$.

Steps:

1. Factor P and Q completely.
2. Record the domain restriction: every root of Q (from the factored form, **before** cancelling).
3. Cancel factors common to numerator and denominator.
4. State the simplified expression **together with** the restrictions.

Output: an equivalent expression in lowest terms, valid on the stated domain.

7.9.3. Arithmetic with rational expressions

All four operations follow the fraction rules from Chapter 2:

- **Multiply:** numerators times numerators, denominators times denominators, then cancel.
- **Divide:** multiply by the reciprocal (and add a restriction: the divisor cannot be zero either).
- **Add/subtract:** rewrite over a **least common denominator** (LCD): the product of every distinct factor of the denominators, each to the highest power it appears.

Worked example (addition).

$$\begin{aligned}\frac{1}{x-1}+\frac{2}{x+1} &= \frac{1\cdot(x+1)}{(x-1)(x+1)}+\frac{2\cdot(x-1)}{(x+1)(x-1)} && \text{(LCD } (x-1)(x+1)\text{)}\\ &= \frac{x+1+2x-2}{(x-1)(x+1)} && \text{(add numerators)}\\ &= \frac{3x-1}{x^2-1}, \qquad x\neq\pm1.\end{aligned}$$

Worked example (multiplication and division).

$$\begin{aligned}\frac{x^2-9}{x^2+x}\div\frac{x-3}{x} &= \frac{x^2-9}{x^2+x}\cdot\frac{x}{x-3} && \text{(multiply by reciprocal)}\\ &= \frac{(x-3)(x+3)}{x(x+1)}\cdot\frac{x}{x-3} && \text{(factor)}\\ &= \frac{x+3}{x+1}, \qquad x\neq 0,\ -1,\ 3. && \text{(cancel } x \text{ and } (x-3)\text{)}\end{aligned}$$

The restrictions come from **every** denominator that ever appeared, including the divisor's numerator $x-3$ (dividing by zero is not allowed).

Back to the cyclist. Setting $T(v)=5$ and multiplying both sides by the LCD $v(v+5)$:

$$\begin{aligned}30(v+5)+30v &= 5v(v+5)\\ 60v+150 &= 5v^2+25v\\ 0 &= 5v^2-35v-150 = 5(v^2-7v-30) = 5(v-10)(v+3).\end{aligned}$$

So $v=10$ km/h (the root $v=-3$ is mathematically valid but physically meaningless). Check: $30/10+30/15=3+2=5$ ✓. Notice that the rational equation became a **quadratic**: the subject of the next chapter.



Cancelling terms instead of factors

$\dfrac{x+6}{x+3}\neq\dfrac{6}{3}=2$ and $\dfrac{x^2+9}{x+3}\neq x+3$. You may only cancel things that **multiply** the whole numerator and the whole denominator. Test with $x=1$: $\frac{7}{4}\neq 2$. A second trap: multiplying an equation by an expression containing the variable can introduce **extraneous solutions** that make an original denominator zero. Always check solutions against the domain.

7.9.4. Where you'd use this

Rational expressions appear whenever rates, shares or reciprocals are combined.

- **Average speed.** Drive to a town at 60 km/h and back at 40 km/h. The average speed is not 50. For a one-way distance d, the total time is $\frac{d}{60}+\frac{d}{40}$, so the average is $\dfrac{2d}{d/60+d/40}=\dfrac{2}{1/60+1/40}=\dfrac{2\cdot60\cdot40}{60+40}=48$ km/h. The d cancels, and simplifying the

general version gives the **harmonic mean** $\frac{2ab}{a+b}$.

- **Physics.** Resistors in parallel combine as $\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$, i.e. $R = \frac{R_1 R_2}{R_1 + R_2}$: 6 Ω and 3 Ω in parallel give $\frac{18}{9} = 2\,\Omega$. The thin-lens equation $\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$ has the same shape.
- **Computing. Amdahl's law** says that if a fraction p of a program's work can be parallelised over n processors, the speed-up is $S(n) = \dfrac{1}{(1-p) + p/n}$. With $p = 0.9$ and 8 cores, $S = 1/(0.1 + 0.1125) \approx 4.7$, not 8, and however many cores you add, S stays below $1/(1-p) = 10$. In machine learning, the **F1 score** that balances precision P and recall R is their harmonic mean $\frac{2PR}{P+R}$: a classifier with $P = 0.8$ and $R = 0.5$ scores $0.8/1.3 \approx 0.62$.

> **US curriculum**
>
> Simplifying and combining rational expressions, and solving rational equations with extraneous-solution checks, are core **Algebra II** topics. Holes and asymptotes of rational **functions** are developed in **Precalculus**.

7.10. Computer-science connections: hashing and checksums

7.10.1. Polynomial rolling hashes

A hash table needs to turn a string into a number. A natural recipe: treat the characters $s_0, s_1, \dots, s_{m-1}$ (as character codes) as the coefficients of a polynomial and evaluate it at some base B, modulo a large prime M to keep the numbers bounded:

$$h(s) = \left(s_0 B^{m-1} + s_1 B^{m-2} + \cdots + s_{m-1} \right) \bmod M.$$

(The notation $a \bmod M$ means the remainder when a is divided by M, Python's `a % M`; see Volume 3, Chapter 23 .) Horner's method computes this in m steps.

The magic is in **sliding** the window. To find a pattern of length m inside a long text, you want the hash of every length-m window. Moving the window one place right removes the leading character and appends a new one:

$$h_{\text{new}} = \left((h_{\text{old}} - s_{\text{out}}\, B^{m-1}) \cdot B + s_{\text{in}} \right) \bmod M.$$

Subtracting the leading term and multiplying by B is exactly the "drop the top coefficient and shift" step of polynomial arithmetic. Each slide costs $O(1)$, so hashing all windows costs $O(n)$ rather than $O(nm)$. This is the **Rabin–Karp** string-search algorithm:

PYTHON

```python
def find_all(text, pattern, base=256, mod=1_000_000_007):
    """Rabin-Karp substring search using a polynomial rolling hash."""
    m = len(pattern)
    if m > len(text):
        return []
    high = pow(base, m - 1, mod)          # weight of the outgoing character

    def h(s):                              # Horner's method, mod `mod`
        v = 0
        for ch in s:
            v = (v * base + ord(ch)) % mod
        return v

    target, window = h(pattern), h(text[:m])
    hits = []
    for i in range(len(text) - m + 1):
        if window == target and text[i:i + m] == pattern:
            hits.append(i)
        if i + m < len(text):              # slide: drop text[i], add text[i+m]
            window = (window - ord(text[i]) * high) % mod
            window = (window * base + ord(text[i + m])) % mod
    return hits

print(find_all("abracadabra", "abra"))
```

```
[0, 7]
```

Why does a polynomial hash work well? Two different strings collide only if the **difference** of their polynomials has B as a root modulo M. A non-zero polynomial of degree less than m has at most $m - 1$ roots (the counting argument from Section 7.6 works modulo a prime too), so if B is chosen at random, the collision probability is at most $(m - 1)/M$: tiny. Backup and deduplication systems use the same idea (Rabin fingerprints) to find chunk boundaries in huge files.

7.10.2. CRCs: polynomial division with XOR

A **cyclic redundancy check** (CRC) protects Ethernet frames, ZIP files and PNG images against corruption. It treats the message bits as the coefficients of a polynomial whose coefficients are just 0 and 1, with arithmetic done modulo 2: $1 + 1 = 0$. In that world addition and subtraction are both XOR, and there are no carries.

The sender and receiver agree on a **generator polynomial** $g(x)$ of degree n, for instance $x^3 + x + 1$ (bits `1011`). To send message $m(x)$:

1. Multiply by x^n (append n zero bits).
2. Divide by $g(x)$ using long division, with XOR for subtraction; keep the remainder $r(x)$ (n bits).
3. Send $m(x)\,x^n + r(x)$ (the message with the remainder appended). Since subtraction is addition mod 2, this is exactly divisible by $g(x)$.

The receiver divides what arrives by $g(x)$: a non-zero remainder means corruption.

PYTHON

```python
def gf2_remainder(bits, gen):
    """Remainder of polynomial division mod 2 (bits highest power first)."""
    bits = list(bits)
    for i in range(len(bits) - len(gen) + 1):
        if bits[i]:                          # leading term present: subtract gen
            for j, g in enumerate(gen):
                bits[i + j] ^= g             # subtraction mod 2 is XOR
    return bits[-(len(gen) - 1):]

msg = [1, 1, 0, 1, 0, 1]
gen = [1, 0, 1, 1]                           # x^3 + x + 1
crc = gf2_remainder(msg + [0, 0, 0], gen)  # multiply by x^3, then divide
sent = msg + crc
print("crc:", crc)
print("check intact:", gf2_remainder(sent, gen))
corrupt = sent.copy()
corrupt[2] ^= 1                              # flip one bit in transit
print("check corrupt:", gf2_remainder(corrupt, gen))
```

```
crc: [1, 1, 1]
check intact: [0, 0, 0]
check corrupt: [1, 0, 1]
```

This is the long-division procedure from Section 7.5 with every subtraction replaced by XOR, which hardware implements with a shift register and a few XOR gates. A flipped bit adds a term x^k to the message polynomial, and the generator is chosen so that no such single term (and no short burst of terms) is divisible by it, so every such error leaves a non-zero remainder.



Polynomials over finite fields

Polynomials with coefficients taken modulo a prime are the backbone of modern coding theory. The QR code on a parcel and the data on a Blu-ray disc use **Reed–Solomon codes**: the data are coefficients of a polynomial, the transmitted symbols are its values at many points, and because a degree-k polynomial is determined by any $k+1$ of its values, the receiver can reconstruct the data even if a large fraction of the symbols are scratched or smudged.



Hash collisions by design

Choose a small modulus M (say 101) and a base B. Type strings and watch their polynomial hashes; the lab searches for two strings with the same hash and shows the difference polynomial having B as a root mod M. Then increase M and watch collisions become rare.

7.11. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Where floating point bites

7.12. Exercises

1. **(Recognition)** For $p(x) = 5 - 3x^4 + x^2 - 7x$, write it in standard form and state the degree, leading coefficient, constant term, and the coefficient of x^3.
2. **(Recognition)** Which of these are polynomials? (a) $3x^2 - \frac{1}{2}x$ (b) $x^{-2} + 1$ (c) $\sqrt{2}\,x^3$ (d) $\sqrt{x} + 1$ (e) $\dfrac{x^2 + 1}{4}$.
3. **(Application)** Expand and simplify: (a) $(3x + 2)(2x^2 - x + 4)$ (b) $(2x - 5)^2$ (c) $(x^2 + 3)(x^2 - 3)$.
4. **(Application)** Compute 103×97 and 51^2 mentally using special products. Show the identity you used.
5. **(Application)** Factor completely: (a) $5x^3 - 20x$ (b) $x^2 - 11x + 28$ (c) $3x^2 + 10x - 8$ (d) $x^3 - 3x^2 - 4x + 12$ (e) $8x^3 + 27$.
6. **(Application)** Divide $x^4 - 3x^3 + 2x - 5$ by $x - 2$ using synthetic division. State the quotient and remainder and confirm the remainder using the remainder theorem.
7. **(Application)** Divide $3x^3 + x^2 - 4$ by $x^2 + 1$ using long division.
8. **(Problem solving)** Find all roots of $x^3 - 7x - 6 = 0$, explaining how you found the first one.
9. **(Problem solving)** Find k so that $x - 2$ is a factor of $x^3 + kx^2 - 4x + 8$.
10. **(Problem solving)** Simplify $\dfrac{x^2 - 5x + 6}{x^2 - 9} \cdot \dfrac{x + 3}{x - 2}$, stating all domain restrictions.
11. **(Problem solving)** Solve $\dfrac{x}{x - 2} - \dfrac{4}{x^2 - 4} = 1$. Watch for extraneous solutions.
12. **(Computing)** Modify `horner` to also return the quotient coefficients, turning it into synthetic division. Test it on $2x^3 - 3x^2 + 5$ at $x = 2$.

▼ **Answers**

1. $p(x) = -3x^4 + x^2 - 7x + 5$. Degree 4, leading coefficient -3, constant term 5, coefficient of x^3 is 0.
2. (a) yes (fractional **coefficients** are fine). (b) no, negative exponent. (c) yes, $\sqrt{2}$ is just a constant coefficient. (d) no, $\sqrt{x} = x^{1/2}$. (e) yes, it is $\frac{1}{4}x^2 + \frac{1}{4}$.
3. (a) $6x^3 - 3x^2 + 12x + 4x^2 - 2x + 8 = 6x^3 + x^2 + 10x + 8$. (b) $4x^2 - 20x + 25$. (c) difference of squares: $x^4 - 9$.
4. $103 \times 97 = (100 + 3)(100 - 3) = 10000 - 9 = 9991$.
   $51^2 = (50 + 1)^2 = 2500 + 100 + 1 = 2601$.
5. (a) $5x(x^2 - 4) = 5x(x - 2)(x + 2)$. (b) need $mn = 28, m + n = -11$: $-4, -7$, so $(x - 4)(x - 7)$. (c) $ac = -24$; $12 \cdot (-2) = -24$, $12 - 2 = 10$: $3x^2 + 12x - 2x - 8 = 3x(x + 4) - 2(x + 4) = (3x - 2)(x + 4)$. (d) grouping:

$x^2(x-3)-4(x-3)=(x-3)(x^2-4)=(x-3)(x-2)(x+2)$. (e) sum of cubes with $a=2x$, $b=3$: $(2x+3)(4x^2-6x+9)$.

6. Coefficients $1,-3,0,2,-5$, $r=2$: 1;\ $1\cdot 2-3=-1$;\ $-1\cdot 2+0=-2$;\ $-2\cdot 2+2=-2$;\ $-2\cdot 2-5=-9$. Quotient x^3-x^2-2x-2, remainder -9. Check: $p(2)=16-24+0+4-5=-9$ ✓.

7. $3x^3\div x^2=3x$; $3x(x^2+1)=3x^3+3x$; subtract: x^2-3x-4. $x^2\div x^2=1$; subtract x^2+1: $-3x-5$. Quotient $3x+1$, remainder $-3x-5$. Check: $(x^2+1)(3x+1)-3x-5=3x^3+x^2+3x+1-3x-5=3x^3+x^2-4$ ✓.

8. Monic, so rational roots divide 6: try $\pm1,\pm2,\pm3,\pm6$. $p(-1)=-1+7-6=0$ ✓. Synthetic division of $1,0,-7,-6$ by -1: $1,-1,-6\mid 0$, quotient $x^2-x-6=(x-3)(x+2)$. Roots $-1,3,-2$.

9. Factor theorem: $p(2)=0$: $8+4k-8+8=0$, so $4k=-8$, $k=-2$.

10. $\dfrac{(x-2)(x-3)}{(x-3)(x+3)}\cdot\dfrac{x+3}{x-2}=1$, with $x\neq 3,-3,2$. The whole expression is the constant 1 with three holes.

11. LCD $(x-2)(x+2)$, restrictions $x\neq\pm2$. Multiply through: $x(x+2)-4=(x-2)(x+2)$, so $x^2+2x-4=x^2-4$, giving $2x=0$, $x=0$. Allowed, and check: $0-4/(-4)=1$ ✓. (No extraneous root here, but you had to check.)

12. Collect the intermediate values: `out = []; r = 0; for c in coeffs: r = r*x + c; out.append(r)`; return `out[:-1], out[-1]`. On `[2, -3, 0, 5]` at 2 the intermediate values are `2, 1, 2, 9`: quotient $2x^2+x+2$, remainder 9, matching the synthetic division example.

7.13. Mastery check

1. State the degree and leading coefficient of $(2x^2 - 1)^3$ without expanding it.
2. Expand $(x - 4)(x^2 + 4x + 16)$ and name the pattern.
3. Factor completely: $2x^4 - 32$.
4. Factor $10x^2 - 11x - 6$.
5. Use the rational root theorem and synthetic division to factor $x^3 + 2x^2 - 5x - 6$.
6. Without dividing, find the remainder when $x^{50} - 3x + 1$ is divided by $x + 1$.
7. Simplify $\dfrac{x^3 - x}{x^2 + 2x + 1}$ and state its domain.
8. Evaluate $3x^4 - 2x^3 + x - 7$ at $x = 2$ with Horner's method, showing each step. How many multiplications did you use?

Decision rule: All eight correct without hesitation → move on to Chapter 8. Missed items 3–5 → rework Section 7.4 and its exercises; factoring fluency is essential for the next chapter. Missed 6 or 8 → reread Section 7.6 and Section 7.8. Missed 7 → reread Section 7.9. Stuck on more than three → work the section exercises first.

▼ **Mastery check answers**

1. Leading term of $(2x^2)^3 = 8x^6$: degree 6, leading coefficient 8.
2. $x^3 + 4x^2 + 16x - 4x^2 - 16x - 64 = x^3 - 64$: difference of cubes $a^3 - b^3 = (a - b)(a^2 + ab + b^2)$ with $a = x$, $b = 4$.
3. $2(x^4 - 16) = 2(x^2 - 4)(x^2 + 4) = 2(x - 2)(x + 2)(x^2 + 4)$.
4. $ac = -60$; need sum -11: $-15 \cdot 4 = -60$, $-15 + 4 = -11$.
   $10x^2 - 15x + 4x - 6 = 5x(2x - 3) + 2(2x - 3) = (2x - 3)(5x + 2)$.
5. Candidates $\pm 1, \pm 2, \pm 3, \pm 6$. $p(-1) = -1 + 2 + 5 - 6 = 0$ ✓. Synthetic division of $1, 2, -5, -6$ by -1: $1, 1, -6 \mid 0$. $x^2 + x - 6 = (x + 3)(x - 2)$. So $(x + 1)(x - 2)(x + 3)$.
6. Remainder theorem with $r = -1$: $(-1)^{50} - 3(-1) + 1 = 1 + 3 + 1 = 5$.
7. $\dfrac{x(x - 1)(x + 1)}{(x + 1)^2} = \dfrac{x(x - 1)}{x + 1}$, domain $x \neq -1$.
8. Coefficients $3, -2, 0, 1, -7$: $3;\ 3 \cdot 2 - 2 = 4;\ 4 \cdot 2 + 0 = 8;\ 8 \cdot 2 + 1 = 17;\ 17 \cdot 2 - 7 = 27$. Value 27, using 4 multiplications (one per degree). Check: $48 - 16 + 2 - 7 = 27$ ✓.

Chapter 8. Quadratics

Chapter map: the quadratic is the first equation in this book that cannot be solved by "undo each operation in turn", and the first function whose graph bends. It is also the best-studied object in elementary algebra, which makes it the ideal place to practise the habits this book cares about: seeing one object as words, symbols, a picture and a computation; knowing several methods and choosing between them; deriving the famous formula rather than memorising it; and knowing exactly when the formula, run on a real computer, gives the wrong answer. Quadratics are also the first non-linear model you meet in practice: anything accelerating, the area of a shape with one adjustable length, any "price times quantity", and any sum of squared errors is one.

Where this chapter sits

You need: solving linear equations (Chapter 5); square roots and exponent laws (Chapter 3); expanding brackets, factoring trinomials, and the factor theorem (Chapter 7). Plotting points in the plane (Chapter 10) helps, but the essentials are recalled here.

This chapter unlocks: Chapter 11 (the quadratic is the model non-linear function, and its transformations are the template for all function transformations); Chapter 15 (what happens when the discriminant is negative); Volume 2, Chapter 18 (optimisation and Newton's method, previewed here); Volume 2, Chapter 32 (floating-point error, of which the quadratic formula is the textbook case); and Volume 2, Chapter 26 (ray tracing, met here as an application).

8.1. Three problems that are secretly the same

8.1.1. A ball thrown upwards

You stand on a platform and throw a ball straight up. It leaves your hand 2 m above the ground, moving upwards at 15 m/s. When does it hit the ground? How high does it go?

If you did sixth-form mechanics, you met the constant-acceleration ("SUVAT") equations. The one that matters here is

$$s = ut + \tfrac{1}{2}at^2,$$

where s is displacement, u initial velocity, a acceleration and t time. Measuring upwards, $u = 15$ m/s and $a = -9.8$ m/s² (gravity pulls down). Adding the 2 m starting height, the height after t seconds is

$$h(t) = 2 + 15t - 4.9t^2.$$

"When does it hit the ground?" becomes: solve $2 + 15t - 4.9t^2 = 0$. The unknown appears both squared and unsquared. You cannot "subtract 2, then divide by 15" as you would for a linear equation, because the t^2 term refuses to go away. You need a new idea.

8.1.2. A fence against a wall

You have 100 m of fencing and want a rectangular pen against a long wall, so you only need to fence three sides. If the two sides perpendicular to the wall are x metres each, the side parallel to the wall is $100 - 2x$, and the area is

$$A(x) = x(100 - 2x) = 100x - 2x^2.$$

Which x gives the biggest pen? Again a squared term, and this time the question is not "where is it zero?" but "where is it largest?"

8.1.3. Breaking even

A small company finds that, selling q hundred units a month, its profit in thousands of pounds is

$$P(q) = -2q^2 + 120q - 1000.$$

At low volume, fixed costs swamp revenue; at very high volume, it has to cut prices so far that it loses money again. For which volumes does it break even ($P = 0$)? Which volume maximises profit?

8.1.4. What they share

All three are about an expression of the form

$$ax^2 + bx + c$$

and all three ask the same two questions: **where is it zero?** and **where is it largest (or smallest)?** By the end of this chapter you will answer both for any such expression, by several methods, and know which method to reach for. (Answers: the ball lands after about 3.19 s and peaks at about 13.48 m; the best pen is 25 m by 50 m; the company breaks even at 1,000 and 5,000 units and does best at 3,000.)

8.2. The vocabulary of quadratics

Before methods, words. Many of these will feel half-familiar; the point of this section is to make each one crisp.



Quadratic expression and quadratic equation

A **quadratic expression** is a polynomial of degree 2: $ax^2 + bx + c$ with $a \neq 0$. A **quadratic equation** is an equation that can be rearranged to

$$ax^2 + bx + c = 0, \qquad a \neq 0,$$

its **standard form**. The word comes from Latin *quadratum*, "a square", because the defining feature is the squared unknown x^2. (Confusingly, *quad* usually means four; here it refers to the four sides of a square.)

Coefficients of a quadratic

In $ax^2 + bx + c$: a is the **leading** (or **quadratic**) coefficient, b the **linear** coefficient, and c the **constant term**. In $7x^2 - 3x + 4$, $a = 7$, $b = -3$, $c = 4$. Always read the signs with the coefficients: in $x^2 - 5x + 6$, $b = -5$, not 5. The condition $a \neq 0$ matters: if $a = 0$ the equation is linear, not quadratic.

Root, zero, solution

A **root** or **solution** of $ax^2 + bx + c = 0$ is a value of x that makes the equation true. The same number is called a **zero** of the function $f(x) = ax^2 + bx + c$, because f takes the value zero there. On a graph, real roots are the **x-intercepts**: where the curve meets the x-axis. A quadratic has at most two roots.

Parabola

The graph of $y = ax^2 + bx + c$ is a **parabola**: a symmetric U-shaped curve. It opens **upwards** (a "cup", $\cup$) if $a > 0$ and **downwards** (a "cap", $\cap$) if $a < 0$. From Greek *parabolē*, "a placing alongside"; the name comes from the ancient Greek classification of conic sections (Chapter 10).

Vertex

The **vertex** of a parabola is its turning point: the lowest point if $a > 0$ (a **minimum**), the highest if $a < 0$ (a **maximum**). For $y = ax^2 + bx + c$ it is at $x = -\dfrac{b}{2a}$. The y-value there is the **minimum value** or **maximum value** of the quadratic.

Axis of symmetry

The **axis of symmetry** is the vertical line through the vertex, $x = -\dfrac{b}{2a}$. The parabola is its own mirror image in this line. Consequence: the two roots (when they exist) sit at equal distances either side of it, so the vertex is exactly halfway between the roots.

Discriminant

The **discriminant** of $ax^2 + bx + c$ is the number

$$\Delta = b^2 - 4ac.$$

(Δ is capital Greek delta; some books write D.) It "discriminates" between the cases: $\Delta > 0$, two distinct real roots; $\Delta = 0$, one repeated real root; $\Delta < 0$, no real roots. Section 8.7 explains why.



Completing the square

Completing the square means rewriting $x^2 + px$ as $\left(x + \frac{p}{2}\right)^2 - \left(\frac{p}{2}\right)^2$, so that the variable appears only once, inside a square. It is the key step in solving any quadratic and in finding its vertex (Section 8.5).

Here is the whole vocabulary on one picture:

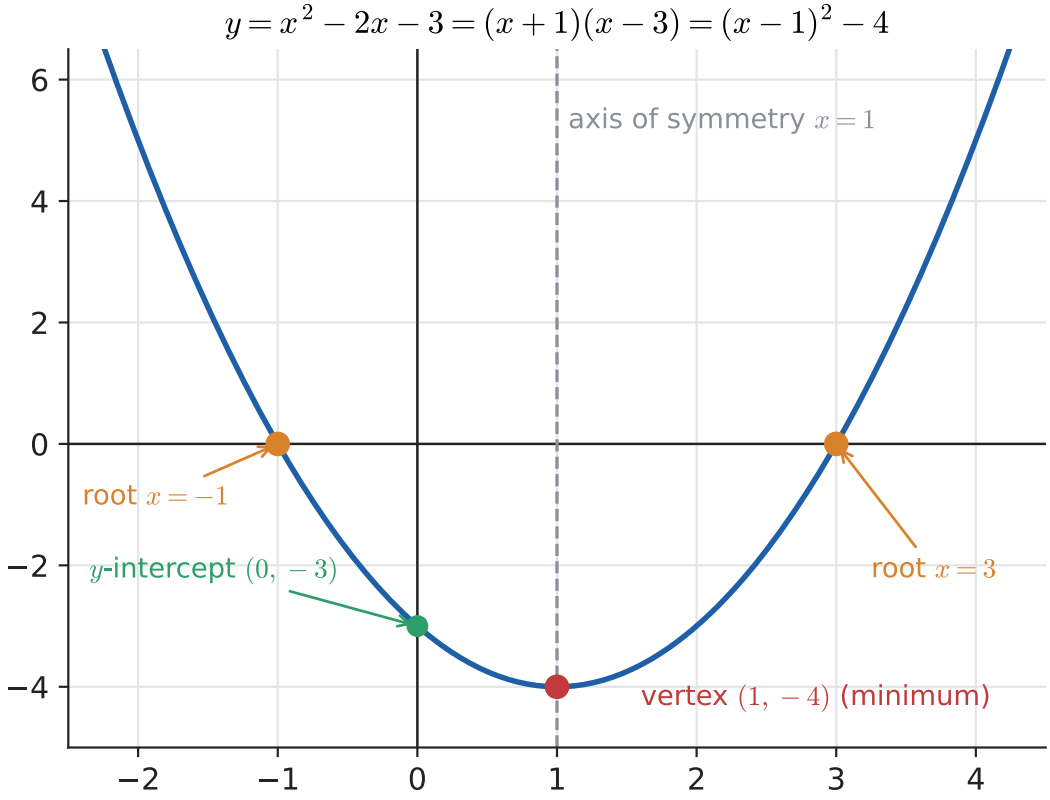


Figure 53. Anatomy of a parabola

Read it carefully, because every label is a fact you can compute from the symbols:

- $a = 1 > 0$, so the parabola opens upwards and the vertex is a minimum.
- The y-intercept is $c = -3$ (set $x = 0$).
- The roots are -1 and 3, because $x^2 - 2x - 3 = (x+1)(x-3)$.
- The axis of symmetry is halfway between the roots, $x = 1$, which agrees with $-b/(2a) = -(-2)/2 = 1$.
- The vertex is $(1, f(1)) = (1, 1 - 2 - 3) = (1, -4)$.
- The discriminant is $(-2)^2 - 4(1)(-3) = 4 + 12 = 16 > 0$: two roots, as seen.

8.3. One object, four representations

The project specification behind this book singles out the quadratic as the example of something that is simultaneously an expression, a function, a curve, a set of roots, a computation and a model. Fluency means moving between these views without friction, so it is worth making each translation explicit

before learning any solution method.

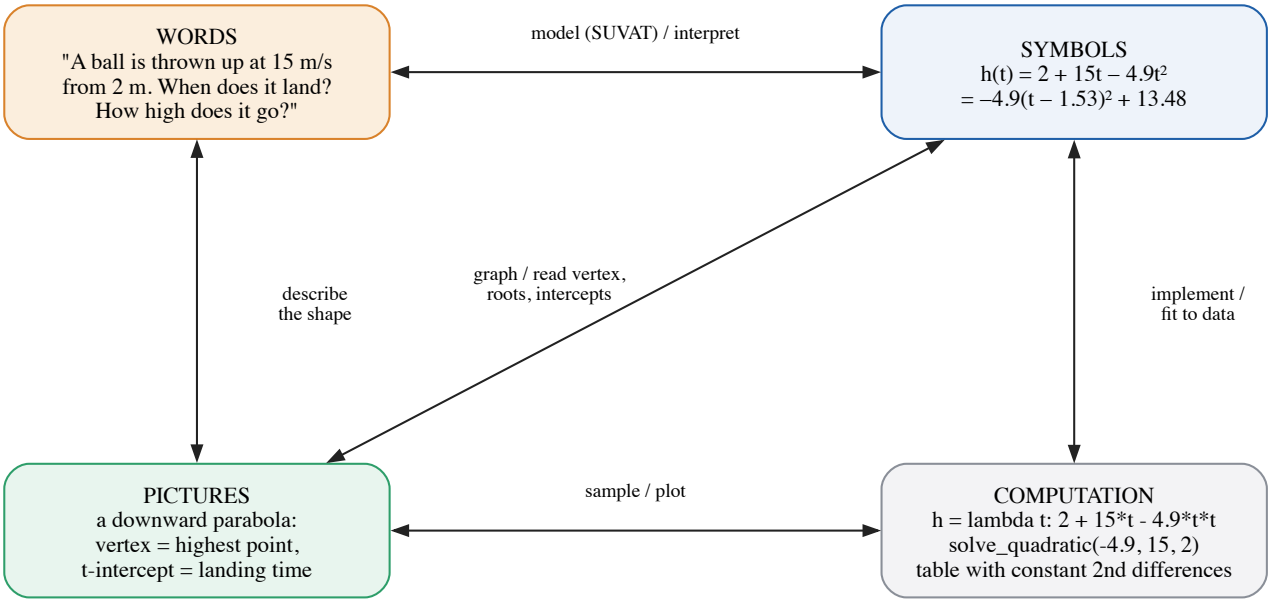


Figure 54. Words, symbols, pictures and computation for the thrown ball

Words → symbols. "Thrown up at 15 m/s from 2 m" becomes $h(t) = 2 + 15t - 4.9t^2$ via the SUVAT equation. Each coefficient has a physical meaning: $c = 2$ is the starting height, $b = 15$ the starting speed, $a = -4.9 = -\frac{1}{2}g$ half the acceleration. Negative a: the parabola opens downwards, so there is a maximum height. The words predicted the shape before you drew anything. *Try it:* Explore the thrown ball's parabola .

Symbols → pictures. Different symbolic forms reveal different features of the graph. Standard form $ax^2 + bx + c$ shows the y-intercept c. Factored form $a(x - r_1)(x - r_2)$ shows the roots. Vertex form $a(x - h)^2 + k$ shows the vertex (h, k) (Section 8.8). For the ball, vertex form is approximately $h(t) = -4.9(t - 1.53)^2 + 13.48$: the peak at $t \approx 1.53$ s, height ≈ 13.48 m, is visible without plotting a single point.

Pictures → words. "Symmetric about its peak" says: the ball takes as long to climb from the release height to the top as to fall back to that height. "Two intercepts, one of them at negative time" says: the equation has a root at negative t that the physics throws away (it is when the ball *would have* been at ground level, had its parabolic flight started earlier).

Symbols ↔ computation. A quadratic is a three-number object (a, b, c) that a computer evaluates with two multiplications and two additions (Horner, Chapter 7). Going the other way, computation reveals a quadratic hiding in data: in a table of values at equally spaced inputs, the **second differences are constant** and equal to $2a$ times the square of the spacing.

```python
def f(x):
    return x**2 - 2*x - 3

xs = list(range(-1, 5))
ys = [f(x) for x in xs]
d1 = [b - a for a, b in zip(ys, ys[1:])]     # first differences
d2 = [b - a for a, b in zip(d1, d1[1:])]     # second differences
print("x :", xs)
print("y :", ys)
print("d1:", d1)
print("d2:", d2)
```

Try it: See the differences table live .

```
x : [-1, 0, 1, 2, 3, 4]
y : [0, -3, -4, -3, 0, 5]
d1: [-3, -1, 1, 3, 5]
d2: [2, 2, 2, 2]
```

Linear functions have constant **first** differences (constant slope, Chapter 5). Quadratics have constant **second** differences: the slope itself changes at a constant rate. That is exactly what constant acceleration means for the ball: velocity changes steadily, so position is quadratic. In Volume 2, Chapter 17 you will see this as "the second derivative of a quadratic is the constant $2a$".



Four views at once

Three sliders for a, b, c. Four synchronised panels: the equation in standard, vertex and factored form (factored form greys out when there are no real roots); the parabola with vertex, axis and roots marked; a table of values with first and second differences; and a short sentence in words ("opens downward, maximum 13.48 at t = 1.53, two real roots"). Drag the graph's vertex directly and watch the coefficients change.

Try it: Open the Four views lab · Parabola with a, b, c sliders

8.4. Method 1: solving by factoring

8.4.1. The problem

Solve $x^2 - 5x + 6 = 0$.

8.4.2. Intuition and why it works

If you can write the quadratic as a product of two linear factors, the zero-product property (Chapter 7) finishes the job: a product is zero exactly when one of its factors is zero. For a monic quadratic $x^2 + bx + c$, expanding $(x + m)(x + n) = x^2 + (m + n)x + mn$ shows you need two numbers with **sum** b and **product** c.

Example 47. Procedure — Solving a quadratic by factoring

Input: a quadratic equation.

Steps:

1. Rearrange to standard form $ax^2 + bx + c = 0$ (everything on one side, zero on the other).
2. Remove any common factor.
3. Factor: for $a = 1$, find m, n with $m + n = b$, $mn = c$; for $a \neq 1$, use the ac method (Chapter 7).
4. Set each factor equal to zero and solve the resulting linear equations.
5. Check each root in the original equation.

Output: the roots, or "does not factor over the integers" (then switch method).

8.4.3. Worked examples

(a) $x^2 - 5x + 6 = 0$. Need sum -5, product 6: -2 and -3.

$$\begin{aligned} x^2 - 5x + 6 &= 0 \\ (x-2)(x-3) &= 0 && ((-2) + (-3) = -5,\ (-2)(-3) = 6) \\ x - 2 = 0 \quad &\text{or} \quad x - 3 = 0 && (\text{zero-product property}) \\ x = 2 \quad &\text{or} \quad x = 3. \end{aligned}$$

Check: $4 - 10 + 6 = 0$ ✓, $9 - 15 + 6 = 0$ ✓.

(b) $2x^2 + x = 6$.

$$\begin{aligned} 2x^2 + x - 6 &= 0 && (\text{standard form}) \\ 2x^2 + 4x - 3x - 6 &= 0 && (ac = -12 = 4 \cdot (-3),\ 4 + (-3) = 1) \\ 2x(x+2) - 3(x+2) &= 0 && (\text{group}) \\ (2x-3)(x+2) &= 0 \\ x = \tfrac{3}{2} \quad &\text{or} \quad x = -2. \end{aligned}$$

(c) The break-even problem: $-2q^2 + 120q - 1000 = 0$.

$$\begin{aligned} -2(q^2 - 60q + 500) &= 0 && (\text{common factor } -2) \\ q^2 - 60q + 500 &= 0 && (\text{divide by } -2) \\ (q-10)(q-50) &= 0 && (\text{sum } -60, \text{ product } 500) \\ q = 10 \quad &\text{or} \quad q = 50. \end{aligned}$$

The company breaks even at 1,000 and 5,000 units (q counts hundreds).

(d) $x^2 = 7x$. Do **not** divide by x:

$$x^2 - 7x = 0 \implies x(x-7) = 0 \implies x = 0 \text{ or } x = 7.$$



Factoring pitfalls

- **Zero on one side first.** $(x-1)(x-2)=6$ does **not** give $x-1=6$ or $x-2=6$. Expand, rearrange to $x^2-3x-4=0$, then factor: $(x-4)(x+1)=0$.
- **Dividing by the variable** loses the root $x=0$, as in example (d).
- **Most quadratics do not factor over the integers.** $x^2-2x-1=0$ has roots $1\pm\sqrt{2}$; no amount of hunting for integer pairs will find them. Factoring is a fast path, not a general method.

When to choose it. When the coefficients are small integers and you can see the factor pair within a few seconds. A quick test: a quadratic with integer coefficients factors over the integers exactly when its discriminant b^2-4ac is a perfect square ($0, 1, 4, 9, 16, \ldots$). For x^2-5x+6, $\Delta=25-24=1=1^2$: factor it.

8.5. Method 2: completing the square

8.5.1. The problem

Solve $x^2+6x-7=0$, but pretend you cannot see the factorisation. More to the point: find a method that works for **every** quadratic, including $x^2-2x-1=0$, which does not factor nicely.

8.5.2. Intuition: equations that are easy

Some quadratics are trivially easy. $x^2=16$ gives $x=\pm4$. Equally easy is

$$(x+3)^2=16 \implies x+3=\pm4 \implies x=1 \text{ or } x=-7.$$

What makes these easy is that x appears **only once**, inside a square, so you can undo the square with a square root and then undo the addition. The general quadratic x^2+6x-7 has x in two places (x^2 and $6x$). Completing the square merges those two appearances into one.

8.5.3. Why it works: literally completing a square

Think of x^2+6x as an area: an x-by-x square plus a rectangle 6 by x. Cut the rectangle into two halves, each 3 by x, and attach one to the right side of the square and the other to the top. You now have an almost-square shape of side $x+3$, missing a 3-by-3 corner.

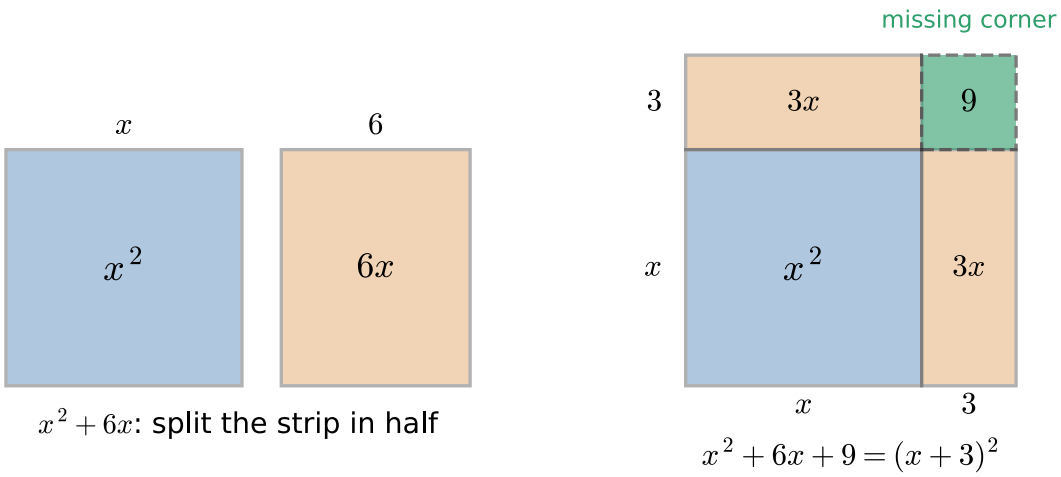


Figure 55. Completing the square: the missing corner has area $(b/2)^2$

Add that corner, area $3^2 = 9$, and you have **completed the square**:

$$x^2 + 6x + 9 = (x + 3)^2.$$

In general, half the coefficient of x, squared, is the missing corner:

$$x^2 + px + \left(\frac{p}{2}\right)^2 = \left(x + \frac{p}{2}\right)^2.$$

(Expand the right side to check: $x^2 + 2 \cdot \frac{p}{2}x + \frac{p^2}{4} = x^2 + px + \frac{p^2}{4}$.) This is the perfect-square identity $(a + b)^2 = a^2 + 2ab + b^2$ from Chapter 7, read backwards.

To keep an **equation** balanced, whatever you add to one side you must add to the other.

Example 48. Procedure — Solving $ax^2 + bx + c = 0$ by completing the square

Input: a, b, c with $a \neq 0$.

Steps:

1. If $a \neq 1$, divide every term by a. Now the equation is $x^2 + px + q = 0$.
2. Move the constant to the right: $x^2 + px = -q$.
3. Add $(p/2)^2$ to **both** sides.
4. Rewrite the left side as $(x + p/2)^2$; simplify the right side to a single number R.
5. If $R < 0$: no real solutions (a square cannot be negative); stop.
6. Take square roots of both sides, **remembering** $\pm$: $x + p/2 = \pm\sqrt{R}$.
7. Subtract $p/2$: $x = -p/2 \pm \sqrt{R}$.

Output: the real roots (two if $R > 0$, one if $R = 0$).

8.5.4. Worked example 1: $x^2 + 6x - 7 = 0$

$$
\begin{aligned}
x^2 + 6x - 7 &= 0 && \\
x^2 + 6x &= 7 && \text{(add 7 to both sides)} \\
x^2 + 6x + 9 &= 7 + 9 && \text{(add } (6/2)^2 = 9 \text{ to both sides)} \\
(x + 3)^2 &= 16 && \text{(left side is a perfect square)} \\
x + 3 &= \pm 4 && \text{(square root, both signs)} \\
x &= -3 \pm 4 && \text{(subtract 3)} \\
x = 1 \quad &\text{or} \quad x = -7. &&
\end{aligned}
$$

Cross-check by factoring: $x^2 + 6x - 7 = (x + 7)(x - 1)$ ✓.

8.5.5. Worked example 2: leading coefficient not 1, irrational roots

Solve $2x^2 - 8x + 3 = 0$.

$$
\begin{aligned}
2x^2 - 8x + 3 &= 0 && \\
x^2 - 4x + \tfrac{3}{2} &= 0 && \text{(divide every term by } a = 2) \\
x^2 - 4x &= -\tfrac{3}{2} && \text{(constant to the right)} \\
x^2 - 4x + 4 &= -\tfrac{3}{2} + 4 && \text{(add } (-4/2)^2 = 4) \\
(x - 2)^2 &= \tfrac{5}{2} && \\
x - 2 &= \pm\sqrt{\tfrac{5}{2}} = \pm\frac{\sqrt{10}}{2} && (\sqrt{5/2} = \sqrt{10}/\sqrt{4}) \\
x &= 2 \pm \frac{\sqrt{10}}{2}. &&
\end{aligned}
$$

Numerically $\sqrt{10}/2 \approx 1.5811$, so $x \approx 3.5811$ or $x \approx 0.4189$. Check the first: $2(3.5811)^2 - 8(3.5811) + 3 \approx 25.649 - 28.649 + 3 = 0.000$ ✓.

8.5.6. Worked example 3: no real roots

Solve $x^2 - 2x + 3 = 0$. *Try it:* See the roots 1 ± i√2 .

$$
\begin{aligned}
x^2 - 2x &= -3 && \\
x^2 - 2x + 1 &= -3 + 1 && \text{(add } (-2/2)^2 = 1) \\
(x - 1)^2 &= -2. &&
\end{aligned}
$$

A real number squared is never negative, so there are no real solutions. Geometrically, the parabola $y = (x - 1)^2 + 2$ has its lowest point at height 2 and never reaches the axis. (In Chapter 15 you will allow $\sqrt{-2} = i\sqrt{2}$, and this equation gets the two roots $1 \pm i\sqrt{2}$.)

8.5.7. Where you'd use this

Completing the square turns "find the best value" into "read off the vertex", because $a(x - h)^2 + k$ is smallest (for $a > 0$) when the square is zero.

- **Statistics: why the mean?** Given data $2, 4, 9$, which single number c is "closest" to all of them, in the sense of making the sum of squared errors $S(c) = (c-2)^2 + (c-4)^2 + (c-9)^2$ smallest? Expanding, $S(c) = 3c^2 - 30c + 101 = 3(c-5)^2 + 26$. The minimum is at $c = 5$, which is exactly the mean, with $S = 26$ (try $c = 4$: $S = 29$). The same algebra with n data points shows that the mean always minimises squared error, which is why least-squares fitting (Volume 3, Chapter 31) and the "mean squared error" loss in machine learning lead back to averages.
- **Optimisation by hand.** The fence and café-revenue problems in Section 8.14 are vertex-reading problems; completing the square is how you find a maximum without calculus.



Completing-the-square slips

- **Add to both sides.** Writing $x^2 + 6x + 9 = 7$ (forgetting to add 9 on the right) is the classic error.
- **Divide by a first.** The "half of p, squared" rule only works when the coefficient of x^2 is 1. With $2x^2 - 8x$, half of -8 squared is 16, which is wrong; dividing by 2 first gives the correct 4.
- **The $\pm$.** $(x+3)^2 = 16$ has two solutions. $\sqrt{16} = 4$ (the square-root symbol always means the non-negative root), but the equation $y^2 = 16$ has solutions $y = \pm 4$.

Why bother, when there is a formula? Three reasons. Completing the square is **how the formula is derived** (next section). It produces **vertex form**, which tells you the maximum or minimum of a quadratic (Section 8.8). And it generalises: you will complete the square again for circles (Chapter 10), in integrals (Volume 2, Chapter 19), and for the Gaussian distribution (Volume 3, Chapter 30).



Al-Khwarizmi's square

The geometric picture above is essentially the one in Muhammad ibn Musa al-Khwarizmi's *The Compendious Book on Calculation by Completion and Balancing* (Baghdad, around 820 CE). His worked problem was "a square and ten roots are equal to thirty-nine", i.e. $x^2 + 10x = 39$; he split the ten into strips of five, completed the square with 25, got $64 = 8^2$, and so $x = 3$. (He ignored the negative root -13; negative numbers were not yet accepted as answers.) The "completion" in his title is *al-jabr*, from which we get the word **algebra**; his own name gives us **algorithm**. Babylonian scribes solved the same kind of problem on clay tablets around 1800 BCE, using recipes that amount to the same steps.

8.6. Method 3: the quadratic formula, derived

8.6.1. The problem

Completing the square works every time, but it means redoing the same steps for each new equation. As programmers know, when you find yourself repeating a procedure with different inputs, you write a function. The quadratic formula is completing the square **done once, symbolically**, for general

a, b, c.

8.6.2. Derivation, every step shown

Start from the general quadratic, with $a \neq 0$:

$$
\begin{aligned}
ax^2 + bx + c &= 0 && \\
x^2 + \frac{b}{a}x + \frac{c}{a} &= 0 && \text{(divide every term by } a\text{; allowed since } a \neq 0\text{)} \\
x^2 + \frac{b}{a}x &= -\frac{c}{a} && \text{(subtract } c/a \text{ from both sides)} \\
x^2 + \frac{b}{a}x + \left(\frac{b}{2a}\right)^2 &= \left(\frac{b}{2a}\right)^2 - \frac{c}{a} && \text{(add half the } x\text{-coefficient, squared, to both sides)} \\
\left(x + \frac{b}{2a}\right)^2 &= \frac{b^2}{4a^2} - \frac{c}{a} && \text{(left side is now a perfect square)} \\
\left(x + \frac{b}{2a}\right)^2 &= \frac{b^2}{4a^2} - \frac{4ac}{4a^2} && \text{(common denominator } 4a^2\text{: multiply } c/a \text{ by } 4a/4a\text{)} \\
\left(x + \frac{b}{2a}\right)^2 &= \frac{b^2 - 4ac}{4a^2} && \text{(combine the fractions)} \\
x + \frac{b}{2a} &= \pm\sqrt{\frac{b^2 - 4ac}{4a^2}} && \text{(square root of both sides; both signs; needs } b^2 - 4ac \geq 0\text{)} \\
x + \frac{b}{2a} &= \pm\frac{\sqrt{b^2 - 4ac}}{2a} && (\sqrt{4a^2} = 2|a|\text{; the } \pm \text{ absorbs the sign of } a\text{)} \\
x &= -\frac{b}{2a} \pm \frac{\sqrt{b^2 - 4ac}}{2a} && \text{(subtract } b/2a \text{ from both sides)} \\
x &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. && \text{(single fraction)}
\end{aligned}
$$

Two steps deserve a comment.

- **Where the discriminant appears.** After completing the square, the right-hand side is $(b^2 - 4ac)/(4a^2)$. The denominator $4a^2$ is always positive, so the sign of the right-hand side is the sign of $b^2 - 4ac$. A square cannot equal a negative number, so if $b^2 - 4ac < 0$ the derivation stops: no real roots. The discriminant is not an arbitrary extra rule; it is the quantity whose square root you need.
- **The $|a|$ subtlety.** $\sqrt{4a^2} = 2|a|$, not $2a$ (if $a = -3$, $\sqrt{36} = 6 = 2|a|$). But since we are taking $\pm$ anyway, $\pm\frac{\sqrt{\Delta}}{2|a|}$ and $\pm\frac{\sqrt{\Delta}}{2a}$ describe the same pair of numbers.

Quadratic formula

The solutions of $ax^2 + bx + c = 0$ ($a \neq 0$) are

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

Read it as "the axis of symmetry $-b/(2a)$, plus or minus a half-width $\sqrt{\Delta}/(2a)$". The two roots are symmetric about the vertex.

That reading is worth pausing on. The formula says the roots are

$$x = \underbrace{-\frac{b}{2a}}_{\text{axis of symmetry}} \pm \underbrace{\frac{\sqrt{b^2 - 4ac}}{2a}}_{\text{distance from axis to each root}},$$

which is exactly what the picture of the parabola shows: two roots, mirror images about the vertical line through the vertex.

Example 49. Procedure — Solving with the quadratic formula

Input: a quadratic equation.

Steps:

1. Rearrange to $ax^2 + bx + c = 0$ and write down a, b, c **with their signs**.
2. Compute the discriminant $\Delta = b^2 - 4ac$.
3. If $\Delta < 0$: no real roots (two complex roots, Chapter 15). Stop.
4. If $\Delta = 0$: one repeated root $x = -b/(2a)$. Stop.
5. Otherwise compute $\sqrt{\Delta}$ and the two roots $x = (-b \pm \sqrt{\Delta})/(2a)$.
6. Simplify surds if an exact answer is wanted; otherwise round only at the end.
7. Check (substitute, or use Vieta's formulas, Section 8.12).

Output: zero, one or two real roots.

8.6.3. Worked example: $3x^2 - 5x - 1 = 0$

$a = 3, b = -5, c = -1$.

$$\Delta = b^2 - 4ac = (-5)^2 - 4(3)(-1) = 25 + 12 = 37 \qquad (>0\text{: two real roots})$$

$$x = \frac{-(-5) \pm \sqrt{37}}{2 \cdot 3} \qquad (\text{substitute; note } -b = +5)$$

$$= \frac{5 \pm \sqrt{37}}{6}.$$

With $\sqrt{37} \approx 6.0828$: $x \approx \frac{11.0828}{6} \approx 1.8471$ or $x \approx \frac{-1.0828}{6} \approx -0.1805$.

Check with Vieta (Section 8.12): sum of roots should be $-b/a = 5/3 \approx 1.6667$; $1.8471 + (-0.1805) = 1.6666$ ✓. Product should be $c/a = -1/3$; $1.8471 \times (-0.1805) \approx -0.3334$ ✓.

8.6.4. Worked example: the ball lands

$-4.9t^2 + 15t + 2 = 0$: $a = -4.9$, $b = 15$, $c = 2$.

$$\Delta = 15^2 - 4(-4.9)(2) = 225 + 39.2 = 264.2, \qquad \sqrt{\Delta} \approx 16.2542$$

$$t = \frac{-15 \pm 16.2542}{2(-4.9)} = \frac{-15 \pm 16.2542}{-9.8}$$

$$t \approx \frac{1.2542}{-9.8} \approx -0.128 \quad \text{or} \quad t \approx \frac{-31.2542}{-9.8} \approx 3.189.$$

The ball hits the ground after about 3.19 s. The negative root is discarded: the model only applies after release. (Discarding roots for physical reasons is normal and is part of the modelling step, not a defect of the mathematics.)

8.6.5. Where you'd use this

The formula is for the quadratics that do not factor nicely, which is nearly all of them in practice.

- **Road safety.** A driver needs 0.7 s to react, then brakes at $6\ \text{m/s}^2$. At speed u m/s the stopping distance is $0.7u + \frac{u^2}{12}$ (thinking distance plus braking distance $u^2/(2 \times 6)$). What is the fastest speed from which the car can stop within 40 m? Multiply $0.7u + \frac{u^2}{12} = 40$ by 12: $u^2 + 8.4u - 480 = 0$, so $u = \frac{-8.4 + \sqrt{8.4^2 + 1920}}{2} = \frac{-8.4 + 44.62}{2} \approx 18.1$ m/s, about 65 km/h. (The negative root is discarded.)
- **Finance: rate of return.** You invest \$1000 and receive \$600 after one year and \$600 after two. The yearly return r that makes this fair satisfies $1000 = \frac{600}{g} + \frac{600}{g^2}$ with $g = 1 + r$. Multiplying by g^2: $1000g^2 - 600g - 600 = 0$, so $g = \frac{600 + \sqrt{600^2 + 4 \cdot 1000 \cdot 600}}{2000} \approx 1.1307$. The **internal rate of return** is about 13.1% a year. With more years of payments the equation becomes a higher-degree polynomial, and spreadsheets solve it numerically.
- **Computing.** Every ray tracer and many collision detectors solve this formula millions of times per second (Section 8.15), which is why its numerical behaviour (Section 8.11) matters.



Quadratic-formula slips

- **Sign of b.** If $b = -5$ then $-b = 5$ and $b^2 = 25$. Type $(-5)^2$, not -5^2 (which calculators and Python read as $-(5^2) = -25$).
- **The fraction bar covers everything.** It is $(-b \pm \sqrt{\Delta})/(2a)$, not $-b \pm \sqrt{\Delta}/(2a)$. In code, write `(-b + d) / (2 * a)`; `-b + d / 2 * a` is wrong twice over.
- **Not in standard form.** For $x^2 = 3x + 4$, first rewrite as $x^2 - 3x - 4 = 0$, so $b = -3$, $c = -4$.
- **On a computer, the formula itself can fail** when $b^2 \gg 4ac$; see Section 8.11.



Did anyone discover "the" formula?

No single person. The Indian mathematician Brahmagupta (628 CE) stated a rule in words equivalent to the formula for one root, and Sridhara (around 900 CE) gave a version that is close to the modern form. The compact symbolic formula needed symbolic algebra itself, which arrived with Viète (1590s) and Descartes (1637). Before that every rule was a recipe written in prose, which is one reason a "procedure" box is such a natural way to present it: that is how it was first written down.

US curriculum

Solving quadratics by factoring, completing the square and the quadratic formula is core **Algebra I** (grades 8–9; Common Core standard A-REI.B.4, which explicitly asks students to **derive** the formula by completing the square). Complex roots and the full discriminant analysis appear in **Algebra II**.

8.7. The discriminant: how many roots, before you find them

8.7.1. The problem

Sometimes you do not need the roots, only to know whether they exist. Does a ray hit a sphere (Section 8.15)? Does a ball thrown at a given speed ever reach a given height? Is a trinomial factorable over the integers? All three are answered by the sign (or the squareness) of one number.

8.7.2. Why it works

The derivation showed

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

The left side is a square, hence ≥ 0. So:

- $\Delta > 0$: the right side is positive and has two square roots, $\pm\sqrt{\Delta}/(2a)$: **two distinct real roots**.

- $\Delta = 0$: the right side is zero, whose only square root is 0: **one repeated root** $x = -b/(2a)$, a root of multiplicity 2 (Chapter 7). The quadratic is a perfect square $a(x + b/2a)^2$.
- $\Delta < 0$: the right side is negative and no real number squares to it: **no real roots** (two complex-conjugate roots).

Graphically, the discriminant measures how far the vertex is from the x-axis, relative to how the parabola opens. The vertex height is $k = -\Delta/(4a)$ (we prove this in Section 8.8). For an upward parabola ($a > 0$), $\Delta > 0$ means $k < 0$: the minimum is below the axis, so the curve must cross it twice.

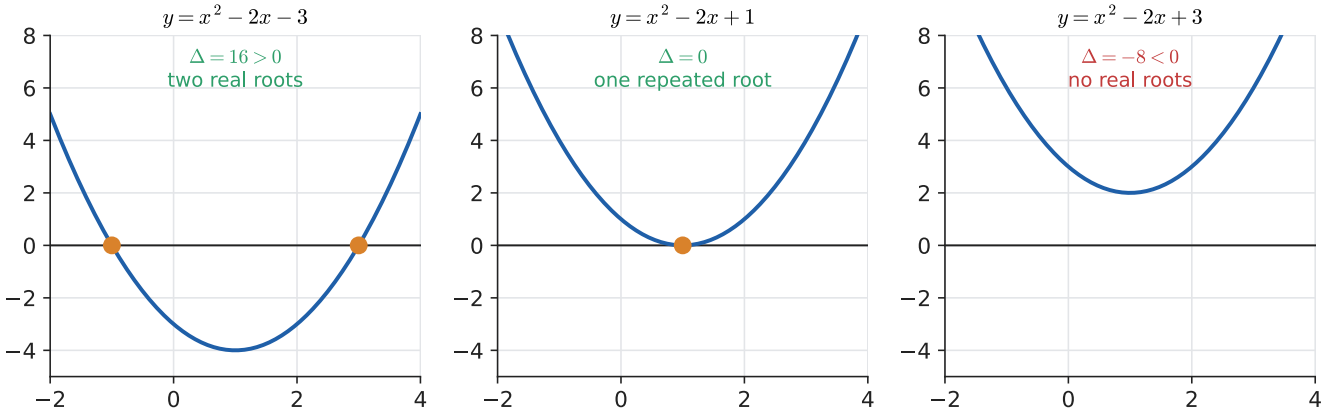


Figure 56. The sign of the discriminant decides how many times the parabola meets the axis

8.7.3. Worked example

How many real roots does each have?

Equation	$\Delta = b^2 - 4ac$	Conclusion
$x^2 - 2x - 3 = 0$	$4 + 12 = 16$	two; and $16 = 4^2$, so it factors: $(x + 1)(x - 3)$
$4x^2 - 12x + 9 = 0$	$144 - 144 = 0$	one repeated: $x = 12/8 = 3/2$; $(2x - 3)^2$
$x^2 - 2x + 3 = 0$	$4 - 12 = -8$	none (real)
$x^2 - 2x - 1 = 0$	$4 + 4 = 8$	two irrational: $1 \pm \sqrt{2}$

Try it: Slide c from Δ = 16 ; See the repeated root 3/2 ; See the complex pair .

Does the ball reach 15 m? Solve $2 + 15t - 4.9t^2 = 15$, i.e. $-4.9t^2 + 15t - 13 = 0$. $\Delta = 225 - 4(-4.9)(-13) = 225 - 254.8 = -29.8 < 0$. No: it never gets that high (consistent with the peak of 13.48 m). **Does it reach 10 m?** $-4.9t^2 + 15t - 8 = 0$ has $\Delta = 225 - 156.8 = 68.2 > 0$: yes, twice, once going up and once coming down.

8.7.4. Where you'd use this

The discriminant answers yes/no questions ("does it ever happen?") without solving anything.

- **Everyday geometry.** Can a rectangle with perimeter 20 m have area 30 m²? Sides x and $10 - x$ give $x(10 - x) = 30$, i.e. $x^2 - 10x + 30 = 0$, with $\Delta = 100 - 120 = -20 < 0$. No: the largest area with that perimeter is the 5 by 5 square, 25 m².
- **Engineering: springs and dampers.** A car's suspension, a door closer and a measuring instrument's needle all behave like a mass m on a spring of stiffness k with damping c. Their motion is governed by the roots of $ms^2 + cs + k = 0$, and the sign of $\Delta = c^2 - 4mk$ decides the character. $\Delta < 0$: it overshoots and oscillates (**underdamped**). $\Delta > 0$: it creeps back slowly (**overdamped**). $\Delta = 0$: it returns as fast as possible without overshooting (**critically damped**). For $m = 1$ kg and $k = 100$ N/m, critical damping needs $c^2 = 400$, i.e. $c = 20$ N·s/m. Engineers tune door closers and instrument needles to sit near that value.
- **Computing.** A ray misses a sphere exactly when the discriminant of its intersection quadratic is negative, so a renderer can reject most rays with one multiply-and-compare and never take a square root (Section 8.15).



Factorability test

A quadratic $ax^2 + bx + c$ with integer coefficients factors into linear factors with integer (or rational) coefficients if and only if its discriminant is a perfect square. Reason: the roots $(-b \pm \sqrt{\Delta})/(2a)$ are rational exactly when $\sqrt{\Delta}$ is.



Discriminant explorer

Fix a and b; slide c. Watch the parabola rise, the two roots slide together, merge at $\Delta = 0$, and vanish. A second panel plots the roots in the **complex plane**: as they leave the real axis they become a symmetric pair $p \pm qi$ moving vertically, preparing the ground for Chapter 15.

Try it: Open the Discriminant explorer · Open the discriminant explorer · Slide c, watch roots merge

8.8. Three forms, and the graph of a quadratic

8.8.1. The problem

Find the maximum height of the ball and when it occurs; find the best fence. These are questions about the **vertex**, not the roots. Each of the three standard ways of writing a quadratic answers a different question directly:

Form	Formula	Reveals
Standard	$y = ax^2 + bx + c$	y-intercept c; coefficients for the formula and discriminant
Factored (intercept)	$y = a(x - r_1)(x - r_2)$	roots r_1, r_2
Vertex	$y = a(x - h)^2 + k$	vertex (h, k), axis $x = h$, max/min value k

In all three, a is the same number and has the same meaning: direction (sign) and steepness (size).

8.8.2. Transformations of $y = x^2$

Vertex form is easiest to understand as a sequence of moves applied to the basic parabola $y = x^2$, whose vertex is the origin.

- $y = x^2 + k$: shift **up** by k (down if $k < 0$).
- $y = (x - h)^2$: shift **right** by h. (The minus sign surprises people. The new curve reaches height 0 when $x - h = 0$, i.e. at $x = h$, so the vertex has moved to $x = h$.)
- $y = ax^2$: stretch vertically by factor a; $|a| > 1$ is narrower, $|a| < 1$ wider; $a < 0$ also reflects in the x-axis.

Combining them: $y = a(x - h)^2 + k$ is $y = x^2$ stretched by a, then moved so that its vertex is at (h, k).

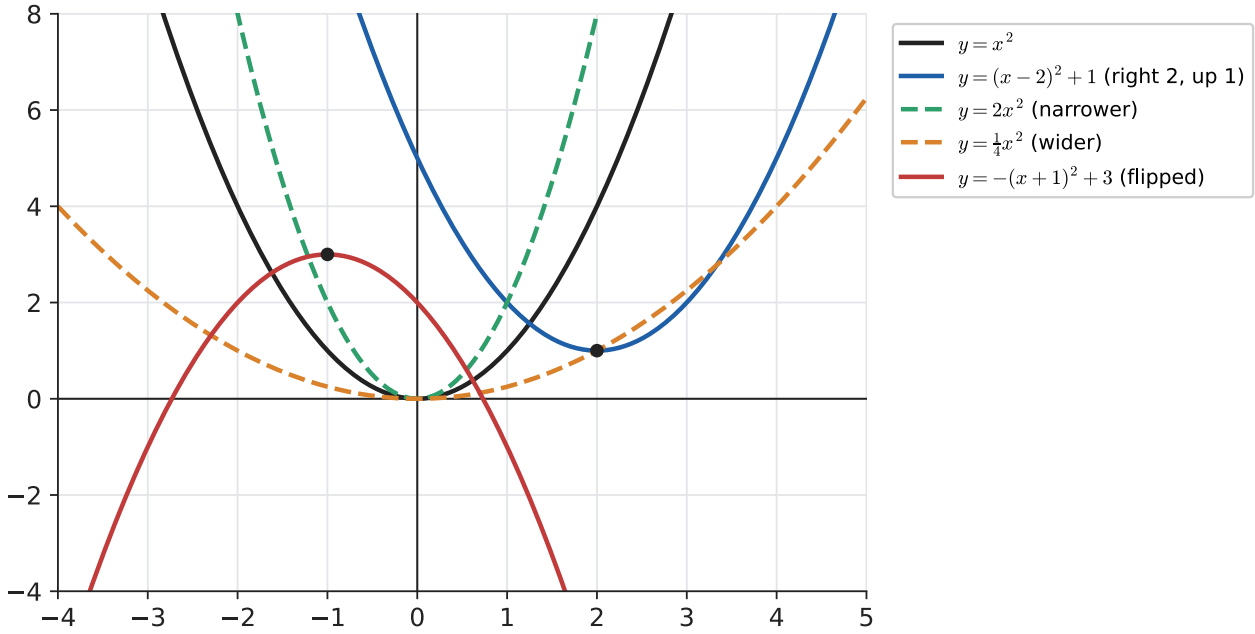


Figure 57. Transformations of the basic parabola

These rules are not special to parabolas. Replacing x by $x - h$ shifts **any** graph right by h, and multiplying by a stretches any graph vertically. You will reuse this exact toolkit for every function family in Chapter 11.

8.8.3. Converting between forms

Standard → vertex: complete the square (keeping the equation as a function, so we add and subtract rather than adding to both sides).

Example 50. Procedure — Standard form to vertex form

Input: $y = ax^2 + bx + c$.

Steps:

1. Factor a out of the first two terms: $y = a\left(x^2 + \frac{b}{a}x\right) + c$.
2. Inside the bracket, add and subtract $\left(\frac{b}{2a}\right)^2$.
3. Write the first three terms inside as a square; move the subtracted term out of the bracket (multiplying it by a).
4. Simplify the constant.

Output: $y = a(x-h)^2 + k$ with $h = -\dfrac{b}{2a}$ and $k = c - \dfrac{b^2}{4a}$.

Worked example. Write $y = 2x^2 - 12x + 13$ in vertex form.

$$
\begin{aligned}
y &= 2(x^2 - 6x) + 13 && \text{(factor 2 from the } x \text{ terms)} \\
&= 2(x^2 - 6x + 9 - 9) + 13 && \text{(add and subtract } (-6/2)^2 = 9) \\
&= 2\big((x-3)^2 - 9\big) + 13 && \text{(perfect square)} \\
&= 2(x-3)^2 - 18 + 13 && \text{(distribute the 2)} \\
&= 2(x-3)^2 - 5.
\end{aligned}
$$

Vertex $(3, -5)$, a minimum because $a = 2 > 0$. The minimum value of $2x^2 - 12x + 13$ is -5, attained at $x = 3$.

The general vertex. Doing the same symbolically:

$$
\begin{aligned}
ax^2 + bx + c &= a\left(x^2 + \frac{b}{a}x\right) + c \\
&= a\left(x^2 + \frac{b}{a}x + \frac{b^2}{4a^2} - \frac{b^2}{4a^2}\right) + c \\
&= a\left(x + \frac{b}{2a}\right)^2 - \frac{b^2}{4a} + c \\
&= a\left(x + \frac{b}{2a}\right)^2 - \frac{b^2 - 4ac}{4a}.
\end{aligned}
$$

So $h = -b/(2a)$ and $k = -\Delta/(4a)$, which is the fact we used in Section 8.7.

Vertex → standard: expand. $2(x-3)^2 - 5 = 2(x^2 - 6x + 9) - 5 = 2x^2 - 12x + 13$.

Standard → factored: find the roots. By any method; then $ax^2 + bx + c = a(x - r_1)(x - r_2)$. For $2x^2 - 12x + 13$: $2(x-3)^2 = 5$ gives $x = 3 \pm \sqrt{5/2}$, so $y = 2\left(x - 3 - \sqrt{5/2}\right)\left(x - 3 + \sqrt{5/2}\right)$. Do not forget the leading a: the factored form of $2x^2 - 12x + 13$ is not $(x - r_1)(x - r_2)$, which has leading coefficient 1.

Factored → vertex: average the roots. The vertex is midway between the roots: $h = (r_1 + r_2)/2$; then $k = f(h)$. For $y = (x+1)(x-3)$: $h = (-1 + 3)/2 = 1$, $k = (2)(-2) = -4$. Vertex $(1, -4)$, matching the anatomy diagram.

8.8.4. Sketching a parabola by hand

Example 51. Procedure — Sketching $y = ax^2 + bx + c$

Steps:

1. Sign of a: cup ($a > 0$) or cap ($a < 0$).
2. y-intercept $(0, c)$.
3. Vertex $h = -b/(2a)$, $k = f(h)$; draw the axis $x = h$.
4. Discriminant: if $\Delta \geq 0$, find the roots and mark them.
5. Reflect the y-intercept in the axis to get a free extra point $(2h, c)$.
6. Draw a smooth symmetric curve through the points.

For the ball: cap; intercept $(0, 2)$; vertex at $t = -15/(2 \cdot (-4.9)) = 15/9.8 \approx 1.531$, $h(1.531) = 2 + 225/19.6 \approx 13.48$; roots ≈ -0.128 and ≈ 3.189; mirror point $(3.061, 2)$.

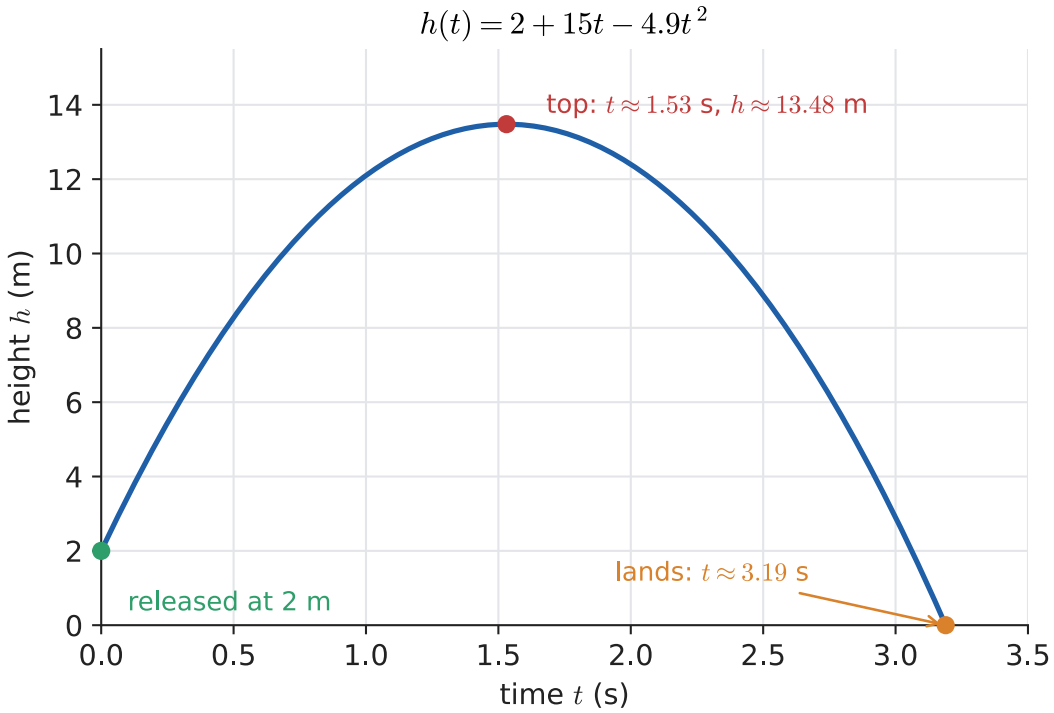


Figure 58. Height of the thrown ball

US curriculum

Graphing quadratics, vertex form and transformations of $y = x^2$ are **Algebra I** topics (Common Core F-IF.C.8a: "use the process of factoring and completing the square in a quadratic function to show zeros, extreme values, and symmetry of the graph"). General function transformations are developed in **Algebra II** and **Precalculus**.

8.9. Method 4 and 5: graphing and numerical root finding

8.9.1. Graphing

Plot $y = ax^2 + bx + c$ and read off where it crosses the axis. On paper this gives one or two significant figures; with graphing software or a zoomable plot, more. Its real value is not precision but **understanding**: a graph shows at a glance how many roots there are, roughly where, and whether the answer from a formula is plausible. Graphing is also the natural method when the "quadratic" is buried in a more complicated equation, such as $x^2 = 2^x$, where you might first want to know how many solutions to look for (there are three).

8.9.2. Numerical root finding

A numerical method does not produce a formula; it produces a sequence of better and better approximations. For a quadratic this is usually unnecessary, because the formula exists. But understanding the numerical approach here, where you can check the answer exactly, prepares you for equations that have no formula: polynomials of degree 5 and above, equations mixing polynomials with e^x or $\sin x$, and the equations inside every physics engine.

Bisection is the simplest. If a continuous function is negative at one point and positive at another, it must cross zero somewhere between (the **intermediate value theorem**, Volume 2, Chapter 16). Repeatedly halve the interval, keeping the half where the sign change is.

Example 52. Procedure — Bisection

Input: continuous f, an interval $[lo, hi]$ with $f(lo)$ and $f(hi)$ of opposite signs, a tolerance ε.

Steps:

1. Compute the midpoint $m = (lo + hi)/2$.
2. If $f(m)$ has the same sign as $f(lo)$, set $lo \leftarrow m$; otherwise set $hi \leftarrow m$.
3. Repeat until $hi - lo < \varepsilon$.

Output: m, within ε of a root. Each step halves the uncertainty, gaining one binary digit of accuracy.

```python
def bisect(f, lo, hi, tol=1e-12):
    """Find a root of f in [lo, hi], assuming f(lo) and f(hi) differ in sign."""
    flo = f(lo)
    steps = 0
    while hi - lo > tol:
        mid = (lo + hi) / 2
        fmid = f(mid)
        if (fmid > 0) == (flo > 0):     # same sign as left end: root is right
            lo, flo = mid, fmid
        else:
            hi = mid
        steps += 1
    return (lo + hi) / 2, steps

root, steps = bisect(lambda x: x*x - 2*x - 1, 2, 3)
print(root, steps)       # exact answer: 1 + sqrt(2) = 2.41421356237...
```

Try it: Run these 40 halvings .

```
2.414213562372879 40
```

Forty halvings to get twelve decimal places: steady but slow.

Newton's method is much faster. Starting from a guess x_0, it repeatedly slides down the tangent line to where the tangent meets the axis:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

where $f'(x)$ is the **derivative**, the slope of the tangent, which for $f(x) = ax^2 + bx + c$ is $2ax + b$. You will derive this properly in Volume 2, Chapter 17 and study Newton's method, and how it can fail, in Volume 2, Chapter 18 . For a preview, take $f(x) = x^2 - 2$ (root $\sqrt{2}$) and $x_0 = 1$: the iteration is $x_{n+1} = x_n - (x_n^2 - 2)/(2x_n)$, giving $1 \to 1.5 \to 1.41\overline{6} \to 1.4142157 \to 1.41421356237469$. The number of correct digits roughly doubles each step: four steps where bisection would need about forty.



Bisection versus Newton

Plot a quadratic and run bisection and Newton side by side, one step per click. Bisection shows a shrinking bracket; Newton shows tangent lines. Choose x_0 exactly at the vertex and watch Newton fail (horizontal tangent, division by zero); start Newton on $x^2 + 1$ and watch it wander chaotically, because there is no real root to find.

Try it: Bisection versus Newton, step by step · Newton on $x^2 + 1$ · Newton from the vertex

8.10. Choosing a method

You now have five ways to solve a quadratic. They are not interchangeable: each has a sweet spot.

A quadratic such as $x^2 - 5x + 6 = 0$ practically begs to be factored: two numbers with sum 5 and product 6 jump out, and $(x - 2)(x - 3) = 0$ is done in seconds with exact answers. Using Newton's method there would be rather silly: you would get 2.0000000000 after several iterations of arithmetic, and you would still have to run it again from a different starting point to find the other root. Conversely, trying to factor $3x^2 - 5x - 1$ wastes time hunting for integer pairs that do not exist.

Method	Use when	Avoid when
Factoring	small integer coefficients; Δ is a perfect square; you want exact roots fast	roots are irrational or the coefficients are large or decimal
Completing the square	you need vertex form (max/min); $a = 1$ and b even; deriving things	messy fractions would appear and you only want roots
Quadratic formula	any quadratic, by hand, when factoring is not obvious; exact surd answers	on a computer when $b^2 \gg 4ac$, unless you use the stable variant
Graphing	you want to **understand**: how many roots, roughly where, sanity checks	you need several significant figures
Numerical (bisection, Newton)	the quadratic is part of a bigger non-polynomial equation; high-degree polynomials; inside an iterative algorithm	a closed form exists and you only need this one quadratic solved

Example 53. Procedure — Method selection for a quadratic equation

Steps:

1. Put it in standard form. If $b = 0$ ($ax^2 + c = 0$), solve directly: $x^2 = -c/a$. If $c = 0$, factor out x.
2. Compute Δ. If $\Delta < 0$ and you only want real roots, stop.
3. If Δ is a perfect square and the coefficients are small, factor.
4. If you also need the vertex, complete the square.
5. Otherwise use the quadratic formula (by hand), or the numerically stable formula (on a computer).
6. Sketch or estimate to sanity-check the answer.

Worked example of choosing. Which method for each?

- $x^2 - 49 = 0$: $b = 0$, so $x = \pm 7$ directly. (Difference of squares, if you prefer.)
- $x^2 + 8x + 11 = 0$: $\Delta = 64 - 44 = 20$, not a square, but b is even and $a = 1$: completing the square is quick: $(x + 4)^2 = 5$, $x = -4 \pm \sqrt{5}$.
- $0.37x^2 - 1.92x + 0.5 = 0$: decimals, so formula (or calculator): $\Delta = 3.6864 - 0.74 = 2.9464$, $x = (1.92 \pm 1.7165)/0.74$, so $x \approx 4.914$ or $x \approx 0.2750$.
- $x^2 = \cos x$: not a quadratic at all; graph to see there are two solutions (near ± 0.82), then Newton.

8.11. When the formula fails: catastrophic cancellation

8.11.1. The problem

The quadratic formula is exact. Floating-point arithmetic is not. Solve *Try it:* See the 25% error .

$$x^2 - 10^8 x + 1 = 0$$

on a computer using the formula exactly as written:

```python
import math

def naive_roots(a, b, c):
    d = math.sqrt(b * b - 4 * a * c)
    return (-b + d) / (2 * a), (-b - d) / (2 * a)

print(naive_roots(1.0, -1e8, 1.0))
```

```
(100000000.0, 7.450580596923828e-09)
```

The roots multiply to $c/a = 1$ (Vieta), so if one is 10^8 the other must be 10^{-8} (to about 16 digits: the exact small root is $1.0000000000000000 \times 10^{-8}$). The computer reports 7.45×10^{-9}: an error of 25%, from a formula you just derived with complete rigour. Nothing is wrong with the algebra. Something is wrong with how the computer carried it out.

8.11.2. Why it happens

A double-precision float carries about 16 significant decimal digits (Volume 2, Chapter 32). Here $b^2 = 10^{16}$ and $4ac = 4$. Then

$$\sqrt{b^2 - 4ac} = \sqrt{10^{16} - 4} \approx 10^8 - 2 \times 10^{-8} = 99\,999\,999.99999998.$$

That number needs 18 significant digits, but doubles near 10^8 are spaced $2^{-26} \approx 1.49 \times 10^{-8}$ apart, so the nearest one the computer can store is $10^8 - 1.49 \times 10^{-8}$. The small root is computed as

$$\frac{-b - \sqrt{\Delta}}{2a} = \frac{10^8 - 99\,999\,999.99999998\ldots}{2},$$

the difference of two nearly equal numbers. Their first 16 digits agree and cancel, leaving only the last one or two digits, which are mostly rounding error: the computer gets $(1.49 \times 10^{-8})/2 \approx 7.45 \times 10^{-9}$, exactly the wrong answer printed above. This is **catastrophic cancellation**: subtraction of nearly equal approximate quantities destroys relative accuracy. The large root, $(-b + \sqrt{\Delta})/(2a)$, **adds** two numbers of the same sign and is perfectly accurate.



Catastrophic cancellation

Catastrophic cancellation is the loss of significant digits when two nearly equal floating-point numbers are subtracted. The subtraction itself is exact; the problem is that the inputs' small rounding errors become large relative to the small result. The cure is almost always algebraic: rewrite the expression so the subtraction never happens.

8.11.3. A by-hand version

You can see the effect without a computer by pretending your calculator keeps only 4 significant digits. Solve $x^2 + 62.10x + 1 = 0$ (true roots ≈ -0.01611 and ≈ -62.08): *Try it:* The 4-digit calculator .

$$\begin{aligned} b^2 &= 3856.41 \to 3856 && \text{(round to 4 digits)} \\ b^2 - 4ac &= 3856 - 4 = 3852 \\ \sqrt{3852} &= 62.064\ldots \to 62.06 \\ x_1 &= \frac{-62.10 + 62.06}{2} = \frac{-0.04}{2} = -0.02000. && \text{(only one significant digit survived)} \end{aligned}$$

The answer -0.02000 is 24% too large in magnitude. The other root, $(-62.10 - 62.06)/2 = -124.16/2 \to -124.2/2 = -62.10$, is off only in its last digit (true value -62.08).

8.11.4. The fix: compute the safe root first, then use Vieta

The product of the roots is $x_1 x_2 = c/a$ (Section 8.12). So compute the root that involves **no cancellation**, and get the other from the product. Concretely, choose the sign in front of the square root to match the sign of b, so that b and $\operatorname{sign}(b)\sqrt{\Delta}$ add rather than cancel:

$$q = -\frac{1}{2}\left(b + \operatorname{sign}(b)\sqrt{b^2 - 4ac}\right), \qquad x_1 = \frac{q}{a}, \qquad x_2 = \frac{c}{q}.$$

Here $\operatorname{sign}(b)$ is $+1$ if $b \geq 0$ and -1 if $b < 0$. Why is c/q a root? Multiply the two roots from the textbook formula: $\frac{-b+\sqrt{\Delta}}{2a} \cdot \frac{-b-\sqrt{\Delta}}{2a} = \frac{b^2-\Delta}{4a^2} = \frac{4ac}{4a^2} = \frac{c}{a}$. So if $x_1 = q/a$ is one root, the other is $\frac{c/a}{q/a} = \frac{c}{q}$.

In the 4-digit example: $62.10 + 62.06 = 124.16 \to 124.2$, so $q = -62.10$, $x_2 = q/a = -62.10$ and $x_1 = c/q = 1/(-62.10) = -0.01610$. The true value is $-0.016107\ldots$, so the small root is now off by at most one unit in the fourth digit, instead of 24%.

```python
import math

def stable_roots(a, b, c):
    d = math.sqrt(b * b - 4 * a * c)
    q = -(b + math.copysign(d, b)) / 2   # b and +-d have the same sign: no cancellation
    return q / a, c / q

print(stable_roots(1.0, -1e8, 1.0))
```

```
(100000000.0, 1e-08)
```

`math.copysign(d, b)` returns d with the sign of b: exactly the $\mathbf{sign}(b)\sqrt{\Delta}$ term.

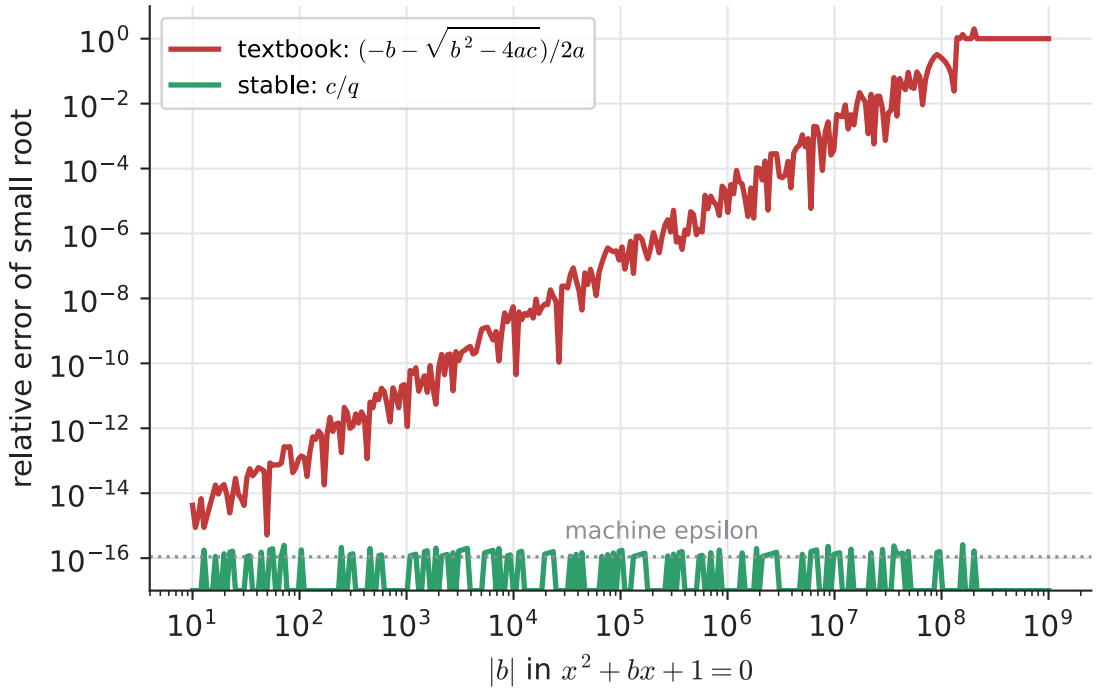


Figure 59. Relative error of the small root of $x^2 + bx + 1 = 0$ as $|b|$ grows

The plot tells the story: the textbook formula loses roughly **two** digits of accuracy for every factor of 10 in $|b|$ (its relative error grows like b^2 times the machine precision $\approx 10^{-16}$, about 25% at $|b| = 10^8$ and 100% from about 1.34×10^8), while the stable version stays at full precision throughout. *Try it:* Slide b and watch digits go .



Formulas are not algorithms

A formula that is exactly correct in real-number arithmetic can be a poor algorithm in floating point. Warning signs: subtracting quantities that may be nearly equal; computing a small number as the difference of two large ones; adding terms of wildly different size. The quadratic formula has a second hazard: b^2 can **overflow** (exceed $\approx 1.8 \times 10^{308}$) even when the roots are modest, so production solvers also rescale the coefficients. Even the discriminant itself suffers cancellation when $b^2 \approx 4ac$ (nearly repeated roots); then the roots are genuinely sensitive to tiny changes in the coefficients, and no rewriting can fully fix that. This is the difference between an unstable **algorithm** (fixable) and an ill-conditioned **problem** (not), a distinction developed in Volume 2, Chapter 32 .

Computer-science connection: a production-quality quadratic solver

Here is a solver that combines the method-selection logic and the stable formula. It treats the degenerate case $a = 0$ (a linear equation) explicitly, distinguishes the three discriminant cases, and never subtracts nearly equal numbers to form a root.

PYTHON

```python
import math

def solve_quadratic(a, b, c):
    """Real roots of a*x**2 + b*x + c = 0, computed stably.

    Returns a sorted tuple of 0, 1 or 2 roots.
    """
    if a == 0:                              # not quadratic: bx + c = 0
        if b == 0:
            raise ValueError("no unknown left to solve for")
        return (-c / b,)
    disc = b * b - 4 * a * c
    if disc < 0:
        return ()                           # complex roots only
    if disc == 0:
        return (-b / (2 * a),)              # repeated root
    q = -0.5 * (b + math.copysign(math.sqrt(disc), b))
    return tuple(sorted((q / a, c / q)))

print(solve_quadratic(1, -5, 6))            # factorable
print(solve_quadratic(1, -1e8, 1))          # cancellation-prone
print(solve_quadratic(1, 2, 1))             # repeated root
print(solve_quadratic(1, 0, 1))             # no real roots
print(solve_quadratic(-4.9, 15, 2))         # the ball
```

```
(2.0, 3.0)
(1e-08, 100000000.0)
(-1.0,)
()
(-0.12798267536970298, 3.1892071651656213)
```

Note that `disc == 0` is an exact floating-point comparison. In practice a computed discriminant is rarely exactly zero, so code that must classify "tangent" cases (collision detection, for example) compares against a tolerance instead; which tolerance is right depends on the scale of the problem.

Real libraries go further. NumPy's `np.roots` does not use the formula at all; it builds a **companion matrix** whose eigenvalues are the roots and calls a general eigenvalue solver (Volume 2, Chapter 28), which works for polynomials of any degree.



How much precision do you lose?

Slide b from 10 to 10^{10} in $x^2 + bx + 1 = 0$. Show both formulas' results digit by digit, with correct digits in green and wrong digits in red, and the number of lost digits plotted against $\log_{10} |b|$. Add a "precision" switch (16, 7, or 4 significant digits) to see the same collapse happen earlier in single precision and in the by-hand 4-digit model.

Try it: How much precision do you lose?



The quadratic formula in the wild

George Forsythe's 1966 essay "How do you solve a quadratic equation?" used exactly this example to argue that numerical analysis is not optional knowledge for anyone who computes. William Kahan, the principal architect of the IEEE 754 floating-point standard, later wrote a careful paper on computing the discriminant itself accurately, using fused multiply–add instructions. The humble quadratic is the classic test case that numerical analysts use to teach the subject.

8.12. Vieta's formulas: what the coefficients know about the roots

8.12.1. The problem

The stable algorithm relied on "the roots multiply to c/a". Where does that come from, and what else is true?

8.12.2. Why it works

If r_1 and r_2 are the roots of $ax^2 + bx + c$, the factor theorem says

$$\begin{aligned} ax^2 + bx + c &= a(x - r_1)(x - r_2) \\ &= a\big(x^2 - (r_1 + r_2)x + r_1 r_2\big) \qquad \text{(expand)} \\ &= ax^2 - a(r_1 + r_2)x + a r_1 r_2. \end{aligned}$$

Two polynomials are equal for every x only if their coefficients match, so

$$b = -a(r_1 + r_2), \qquad c = a\, r_1 r_2.$$



Vieta's formulas

For $ax^2 + bx + c = 0$ with roots r_1, r_2 (real or complex, counted with multiplicity):

$$r_1 + r_2 = -\frac{b}{a}, \qquad r_1 r_2 = \frac{c}{a}.$$

For a monic quadratic $x^2 + bx + c$: the roots **sum** to $-b$ and **multiply** to c. This is exactly why factoring $x^2 + bx + c$ means "find two numbers with sum b and product c" (the numbers are the **negatives** of the roots).

8.12.3. Uses

- **Checking answers** without substituting (as in the formula example above).
- **Building a quadratic from its roots.** Roots $3 \pm \sqrt{2}$: sum 6, product $9 - 2 = 7$, so $x^2 - 6x + 7 = 0$.
- **The other root for free.** $x = 2$ is a root of $3x^2 - 4x - 4 = 0$; the product of roots is $-4/3$, so the other root is $-\frac{4}{3} \div 2 = -\frac{2}{3}$.
- **Stable computation**, as in Section 8.11.
- **Symmetric expressions in the roots without finding them.** For $x^2 - 5x + 3 = 0$, $r_1^2 + r_2^2 = (r_1 + r_2)^2 - 2r_1 r_2 = 25 - 6 = 19$.



> *Viète*
> François Viète (1540–1603), a French lawyer and royal cryptanalyst who broke Spanish ciphers for Henri IV, was among the first to use letters systematically for **known** quantities as well as unknowns, which is what made general formulas like these expressible at all. His relations extend to every degree: for a cubic $x^3 + bx^2 + cx + d$ with roots r_1, r_2, r_3, the roots sum to $-b$, their pairwise products sum to c, and their product is $-d$.

8.13. Quadratic inequalities

8.13.1. The problem

For which times is the ball above 10 m? For which production levels is the company profitable? These ask where a quadratic is **positive** (or negative), not where it is zero: inequalities such as $-2q^2 + 120q - 1000 > 0$.

8.13.2. Intuition

A parabola can only change sign where it crosses the axis, at its roots. Between consecutive roots its sign is constant. So: find the roots, then decide the sign on each interval. For an upward parabola ($a > 0$) with two roots, the quadratic is **negative between the roots and positive outside them**; for a downward one, the reverse.

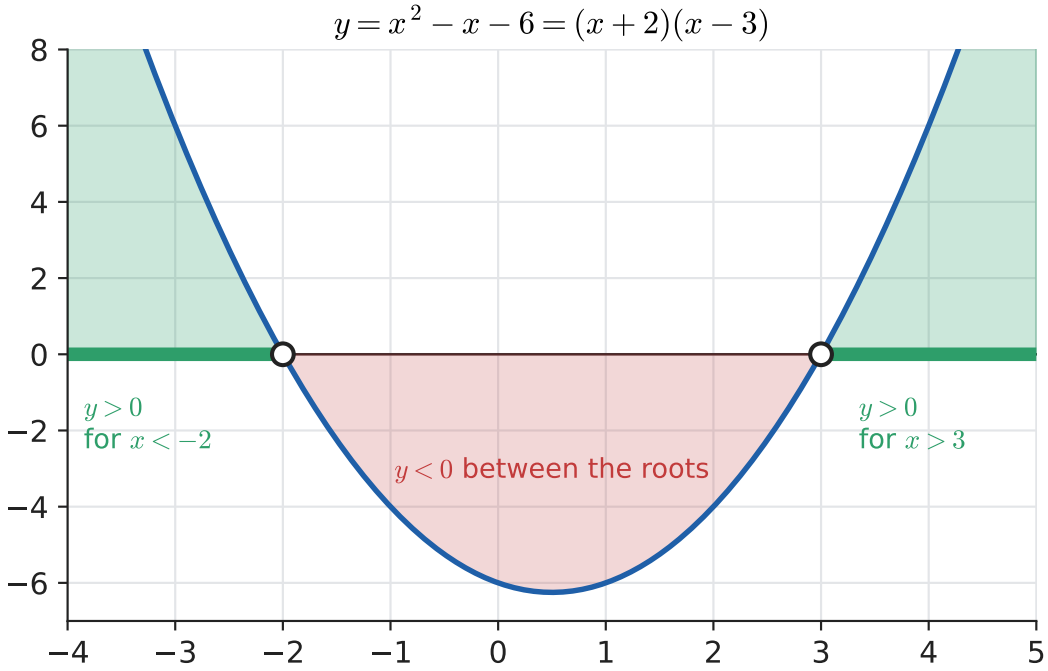


Figure 60. A quadratic is negative between its roots and positive outside them (for $a > 0$)

Example 54. Procedure — Solving a quadratic inequality

Input: an inequality involving a quadratic.

Steps:

1. Rearrange to $ax^2 + bx + c > 0$ (or $<, \geq, \leq$): zero on one side.
2. Find the real roots (any method). If there are none, the quadratic has the sign of a everywhere: the solution is "all real x" or "no x".
3. Sketch the parabola (direction from a; crossings at the roots), or make a sign table using test points.
4. Read off the intervals where the sign is as required. Include the roots for $\geq$ or $\leq$; exclude them for $>$ or $<$.

Output: the solution set, as inequalities or interval notation.

8.13.3. Worked examples

(a) $x^2 - x - 6 > 0$. *Try it:* Sign chart for x² − x − 6 .

$$x^2 - x - 6 = (x+2)(x-3), \qquad \text{roots } -2,\ 3.$$

$a = 1 > 0$, so the parabola is above the axis outside the roots: $x < -2$ or $x > 3$. A sign table confirms:

Interval	$x+2$	$x-3$	product
$x<-2$ (test -3)	$-$	$-$	$+$
$-2<x<3$ (test 0)	$+$	$-$	$-$
$x>3$ (test 4)	$+$	$+$	$+$

In **interval notation** the answer is $(-\infty,-2)\cup(3,\infty)$. Here round brackets mean the endpoint is excluded, square brackets would mean included, ∞ is not a number but a marker for "no upper end", and $\cup$ (union) means "or".

(b) $-x^2+4x+5\geq 0$. Multiply by -1 to make the leading coefficient positive, **reversing the inequality**: *Try it:* Sign chart for example (b) .

$$\begin{aligned} x^2-4x-5 &\leq 0 && \text{(multiply by } -1\text{; flip } \geq \text{ to } \leq\text{)}\\ (x-5)(x+1) &\leq 0 && \text{(roots } -1,\ 5\text{)}\\ -1\leq x &\leq 5. && \text{(between the roots, endpoints included)} \end{aligned}$$

(c) Profitable production: $-2q^2+120q-1000>0$ is $q^2-60q+500<0$ (divide by -2, flip), i.e. $(q-10)(q-50)<0$, so $10<q<50$: between 1,000 and 5,000 units. *Try it:* Sign chart for the profit .

(d) The ball above 10 m: $-4.9t^2+15t-8>0$. Roots (formula, $\Delta=68.2$, $\sqrt{\Delta}\approx 8.2583$): $t=(15\mp 8.2583)/9.8\approx 0.688$ and 2.373. Downward parabola, so positive between the roots: about $0.69<t<2.37$ seconds, a window of $\sqrt{\Delta}/|a|\approx 1.69$ s. *Try it:* When is the ball above 10 m? .



Inequality traps

- **Square-rooting an inequality.** $x^2>4$ does **not** mean $x>2$; it means $x>2$ **or** $x<-2$ (try $x=-5$). Always go via the roots and a sketch.
- **Multiplying or dividing by a negative number flips the inequality.** Dividing by an expression whose sign you do not know is forbidden: you cannot tell whether to flip.
- **No real roots does not mean no solution.** $x^2+1>0$ is true for every x; $x^2+1<0$ for none.

US curriculum

Quadratic inequalities and sign analysis usually appear in **Algebra II**, and again in **Precalculus** for polynomial and rational inequalities. Interval notation is typically introduced in Algebra II.

Try it: Try a rational inequality .

8.14. Applications

8.14.1. Projectile motion

Every object moving under constant acceleration (and nothing else) has a position that is a quadratic function of time. That includes a thrown ball, a dropped phone, a braking car (constant deceleration), and a rocket in the short phase where thrust and gravity are roughly constant.

Worked example: maximum height. For $h(t) = 2 + 15t - 4.9t^2$, the vertex is at

$$t^* = -\frac{b}{2a} = -\frac{15}{2(-4.9)} = \frac{15}{9.8} \approx 1.531 \text{ s},$$

and the maximum height is

$$h(t^*) = c - \frac{b^2}{4a} = 2 - \frac{225}{4(-4.9)} = 2 + \frac{225}{19.6} \approx 2 + 11.480 = 13.48 \text{ m}.$$

In SUVAT language you would have said: at the top the velocity is zero, $v = u + at$ gives $0 = 15 - 9.8t$, so $t = 15/9.8$. Same answer, reached through physics instead of algebra, and a good illustration that the vertex of a quadratic is where its rate of change is zero, the idea that drives Volume 2, Chapter 18 .

Braking distance. A car travelling at u m/s decelerates at d m/s². Its stopping distance is $u^2/(2d)$: quadratic in speed. Doubling speed **quadruples** braking distance, which is why the UK Highway Code's stopping distances grow so steeply.

8.14.2. Optimising area and revenue

Any quantity that is a product of two things, one increasing and one decreasing linearly in the same variable, is a downward quadratic and has a maximum at its vertex.

The fence. $A(x) = x(100 - 2x) = -2x^2 + 100x$. Vertex at $x = -100/(2 \cdot (-2)) = 25$. Area $A(25) = 25 \times 50 = 1250$ m². Shortcut: the roots are 0 and 50, and the vertex is midway between them.

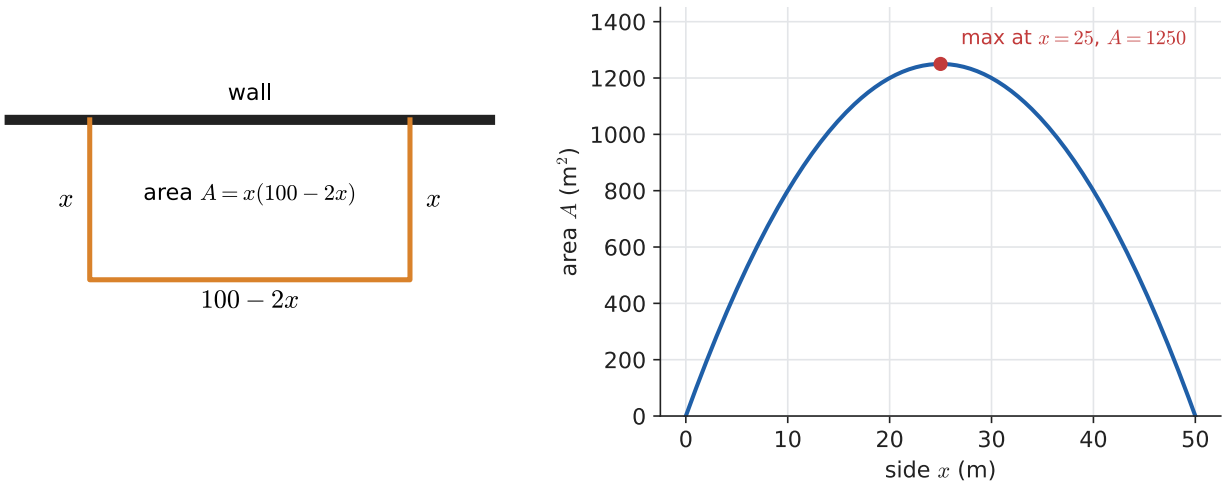


Figure 61. The pen against a wall and its area function

Revenue. A café's simple demand model says it would sell 1000 coffees a week if they were free, and loses 20 weekly sales for every £1 on the price: demand $q = 1000 - 20p$. Revenue is price times quantity:

$$R(p) = p(1000 - 20p) = -20p^2 + 1000p.$$

Roots $p = 0$ and $p = 50$; vertex midway at $p = 25$, with $R(25) = 25 \times 500 = 12{,}500$, i.e. £12,500 a week. Raising the price beyond £25 loses more in volume than it gains in margin. (Real demand curves are not linear, but this is the first model economists draw, and the reasoning is the same with a better model plus calculus.)

The break-even company is at its best at the midpoint of its break-even volumes, $q = 30$: $P(30) = -1800 + 3600 - 1000 = 800$, i.e. £800,000 at 3,000 units.

8.14.3. The reflective property: dishes, headlights and telescopes

Every parabola has a special point, its **focus**. For $y = \dfrac{x^2}{4f}$ the focus is at $(0, f)$. Every ray travelling parallel to the axis that hits the inside of the parabola reflects through the focus. Spin the parabola about its axis to get a **paraboloid**, and every parallel ray striking the dish converges at one point. *Try it:* Rays meeting at the focus .

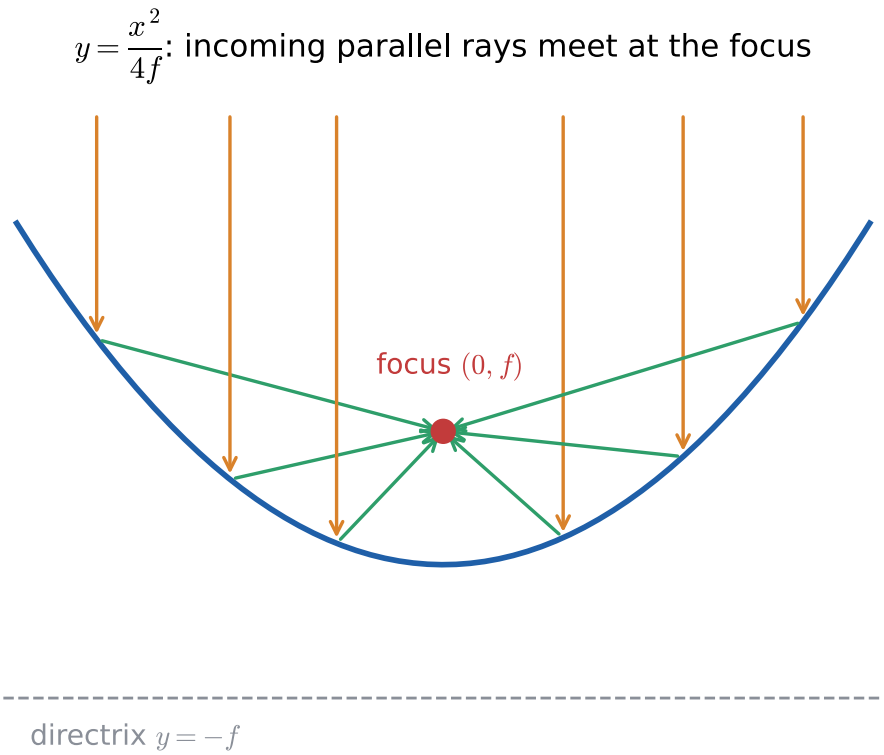


Figure 62. Rays parallel to the axis all reflect through the focus

That is why:

- **Satellite dishes** put the receiver at the focus: the faint parallel signal from a satellite 36,000 km away is concentrated there.
- **Reflecting telescopes** (Newton's design and every large modern telescope) use paraboloidal mirrors to bring starlight to a focus.
- **Headlights and torches** run the reflection backwards: a bulb at the focus sends light out in a parallel beam.
- **Solar cookers** and solar-thermal power plants concentrate sunlight on a pipe or pot at the focus.

Worked example: where to put the receiver. A dish is 60 cm across and 9 cm deep. Place its vertex at the origin, so its cross-section is $y = x^2/(4f)$ and the rim is at $(30, 9)$: *Try it:* Place the receiver yourself .

$$\begin{aligned} 9 &= \frac{30^2}{4f} && \text{(rim lies on the curve)} \\ 36f &= 900 && \text{(multiply both sides by } 4f\text{)} \\ f &= 25 \text{ cm}. \end{aligned}$$

The receiver sits 25 cm in front of the centre of the dish. The reason the property holds is a short calculation with distances: every point of the parabola is equally far from the focus and from a line called the **directrix** ($y = -f$); the proof uses tangent lines and is done in Chapter 10.



Whispering galleries and burning mirrors

A legend says Archimedes set fire to Roman ships at Syracuse (212 BCE) with "burning mirrors". Whether or not it happened (modern attempts have produced mixed results), Diocles' treatise *On Burning Mirrors* (around 190 BCE) contains the first known proof of the focal property of the parabola. The same geometry, applied to the ellipse, makes "whispering galleries" where a whisper at one focus is heard clearly at the other.

8.15. Computer-science connection: ray–sphere intersection

8.15.1. The problem

A ray tracer makes images by firing a ray from the camera through each pixel and asking: **what does this ray hit first?** For a sphere, that question is a quadratic equation, solved millions of times per frame. This is perhaps the most-executed quadratic formula in the world.

8.15.2. Setting it up

A **ray** starts at a point $\mathbf{o}$ (the camera) and travels in direction $\mathbf{d}$. Bold letters denote **vectors**, here just triples of numbers (x, y, z) (Volume 2, Chapter 26). The points on the ray are

$$\mathbf{p}(t) = \mathbf{o} + t\,\mathbf{d}, \qquad t \geq 0,$$

meaning: start at $\mathbf{o}$ and move t units of $\mathbf{d}$; component by component, $p_x = o_x + t d_x$ and so on. The parameter t measures how far along the ray you are; smaller t is closer to the camera.

A **sphere** with centre $\mathbf{c}$ and radius r is the set of points at distance r from $\mathbf{c}$. Squared distance is computed by Pythagoras in 3D: for a vector $\mathbf{v} = (v_x, v_y, v_z)$,

$$\mathbf{v} \cdot \mathbf{v} = v_x^2 + v_y^2 + v_z^2,$$

the **dot product** of $\mathbf{v}$ with itself. More generally, $\mathbf{u} \cdot \mathbf{v} = u_x v_x + u_y v_y + u_z v_z$; it distributes over addition just like ordinary multiplication.

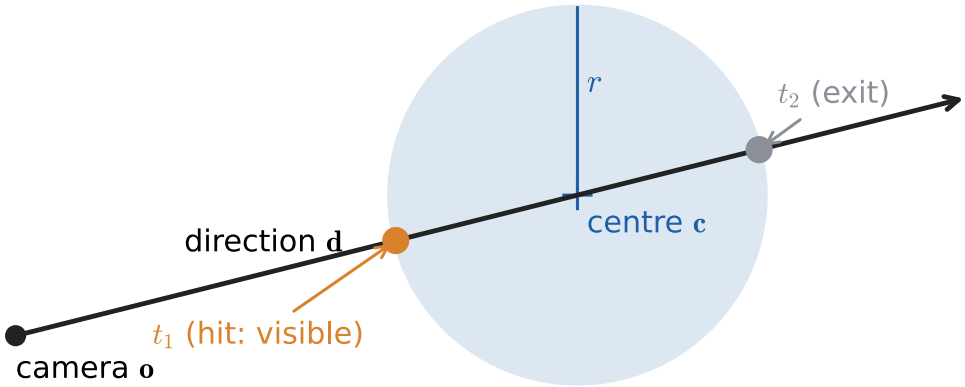


Figure 63. A ray hitting a sphere: the two values of t are where the ray enters and leaves

8.15.3. Deriving the quadratic

The ray point $\mathbf{p}(t)$ is on the sphere when its squared distance from the centre is r^2. Write $\mathbf{f} = \mathbf{o} - \mathbf{c}$ for the vector from the centre to the camera. Then $\mathbf{p}(t) - \mathbf{c} = \mathbf{f} + t\mathbf{d}$, and

$$\begin{aligned}
(\mathbf{f} + t\mathbf{d}) \cdot (\mathbf{f} + t\mathbf{d}) &= r^2 && \text{(squared distance equals } r^2\text{)} \\
\mathbf{f}\cdot\mathbf{f} + 2t\,(\mathbf{d}\cdot\mathbf{f}) + t^2\,(\mathbf{d}\cdot\mathbf{d}) &= r^2 && \text{(expand, like } (f+td)^2\text{)} \\
\underbrace{(\mathbf{d}\cdot\mathbf{d})}_{a}\,t^2 + \underbrace{2(\mathbf{d}\cdot\mathbf{f})}_{b}\,t + \underbrace{(\mathbf{f}\cdot\mathbf{f} - r^2)}_{c} &= 0. && \text{(standard form in } t\text{)}
\end{aligned}$$

A quadratic in t, and every piece of this chapter now has a geometric meaning:

- $\Delta < 0$: the ray **misses** the sphere. (This is the common case, and you can decide it without a square root.)
- $\Delta = 0$: the ray grazes the sphere tangentially.
- $\Delta > 0$: two intersections, $t_1 < t_2$: entering and leaving. The visible surface is at the smallest **positive** t.
- $t_1 < 0 < t_2$: the camera is **inside** the sphere.
- Both negative: the sphere is behind the camera.

8.15.4. Worked example

Camera at $\mathbf{o} = (0, 1, 0)$, looking along $\mathbf{d} = (1, 0, 0)$; sphere centre $\mathbf{c} = (5, 0, 0)$, radius $r = 2$.

$$\begin{aligned}
\mathbf{f} &= \mathbf{o} - \mathbf{c} = (-5, 1, 0) \\
a &= \mathbf{d}\cdot\mathbf{d} = 1 \\
b &= 2(\mathbf{d}\cdot\mathbf{f}) = 2(1\cdot(-5) + 0\cdot 1 + 0\cdot 0) = -10 \\
c &= \mathbf{f}\cdot\mathbf{f} - r^2 = 25 + 1 + 0 - 4 = 22 \\
\Delta &= b^2 - 4ac = 100 - 88 = 12 > 0 && \text{(hit)} \\
t &= \frac{10 \pm \sqrt{12}}{2} = 5 \pm \sqrt{3} \approx 3.268 \text{ or } 6.732.
\end{aligned}$$

The visible point is at $t \approx 3.268$: $\mathbf{p} \approx (3.268, 1, 0)$. Check: its distance from the centre is $\sqrt{(3.268-5)^2 + 1^2} = \sqrt{3.000 + 1} = 2$ ✓.

PYTHON

```python
import math

def dot(u, v):
    return sum(ui * vi for ui, vi in zip(u, v))

def ray_sphere(origin, direction, centre, radius, eps=1e-9):
    """Smallest t > eps where origin + t*direction meets the sphere, or None."""
    f = [o - c for o, c in zip(origin, centre)]   # centre -> camera
    a = dot(direction, direction)
    b = 2 * dot(direction, f)
    c = dot(f, f) - radius * radius
    disc = b * b - 4 * a * c
    if disc < 0:
        return None                               # ray misses
    q = -0.5 * (b + math.copysign(math.sqrt(disc), b))   # stable formula
    if q == 0:
        return None                               # degenerate: grazing at origin
    t1, t2 = sorted((q / a, c / q))
    if t1 > eps:
        return t1                                 # first hit in front of camera
    if t2 > eps:
        return t2                                 # camera is inside the sphere
    return None                                   # sphere is behind the camera

print(ray_sphere((0, 1, 0), (1, 0, 0), (5, 0, 0), 2))   # hit
print(ray_sphere((0, 3, 0), (1, 0, 0), (5, 0, 0), 2))   # passes above: miss
print(ray_sphere((5, 0, 0), (1, 0, 0), (5, 0, 0), 2))   # from inside
```

```
3.267949192431123
None
2.0
```

The small `eps` guards against a classic ray-tracing bug, "shadow acne": a ray bounced off a surface, starting **on** that surface, finds a spurious intersection at $t \approx 10^{-15}$ due to rounding, so the surface shadows itself. The stable formula matters here too, though not where you might guess. A small sphere far away is harmless: both formulas agree to about 16 digits. The textbook formula loses digits when c is tiny compared with b^2, which happens when the ray starts very close to the surface of a large sphere (a shadow or reflection ray leaving a big floor or planet). Then the near root comes from cancellation, which is exactly the shadow-acne situation above.



A one-sphere ray tracer

A 200 by 200 pixel canvas. For each pixel, fire a ray and colour it by the discriminant: black for $\Delta < 0$, and shaded by t_1 (near is bright) otherwise. Move the camera into the sphere and watch the $t_1 < 0 < t_2$ case take over. Then toggle "textbook formula" and push the sphere far away to see cancellation noise appear at the edges.

US curriculum

Ray–sphere intersection is not school mathematics; it uses **Algebra I** quadratics with vectors from **Precalculus** (or a college linear-algebra course). It is a common first project in university computer-graphics courses.

8.16. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Algebra step by step
- Graphing calculator
- Complex numbers, visually
- Root finding, step by step
- Where floating point bites
- Conics, pixels and metrics

8.17. Exercises

1. **(Recognition)** For $5 - 3x^2 + 2x = 0$, identify a, b, c and compute the discriminant. How many real roots?
2. **(Recognition)** Without solving, state the vertex, the axis of symmetry and whether the vertex is a maximum or minimum: (a) $y = -3(x + 2)^2 + 7$ (b) $y = (x - 4)(x + 6)$.
3. **(Application)** Solve by factoring: (a) $x^2 + x - 20 = 0$ (b) $3x^2 - 10x - 8 = 0$ (c) $4x^2 = 9$.
4. **(Application)** Solve by completing the square, giving exact answers: (a) $x^2 - 8x + 3 = 0$ (b) $3x^2 + 12x - 1 = 0$.
5. **(Application)** Solve with the quadratic formula, giving exact and 4-significant-figure answers: $2x^2 + 3x - 7 = 0$.
6. **(Application)** Write $y = -x^2 + 6x - 5$ in vertex form and in factored form, and sketch it.
7. **(Application)** Solve $2x^2 - 5x - 3 \le 0$ and $x^2 + 4x + 5 > 0$.
8. **(Problem solving)** Find all values of k for which $x^2 + kx + 9 = 0$ has exactly one real root. For which k does it have no real roots?
9. **(Problem solving)** A stone is thrown upward at 20 m/s from a cliff top 25 m above the sea (take $g = 10$ m/s²). Find when it hits the water and its greatest height above the sea.
10. **(Problem solving)** A rectangle has perimeter 36 cm and area 72 cm². Find its dimensions using Vieta's formulas (set up a quadratic whose roots are the side lengths).

11. **(Problem solving)** Derive the formula for the x-coordinate of the vertex, $-b/(2a)$, from the quadratic formula by averaging the two roots.

12. **(Computing)** Using the `naive_roots` and `stable_roots` functions, solve $x^2 + 10^5 x + 1 = 0$ and $x^2 - 10^{12} x + 1 = 0$. How many correct significant digits does the naive small root have in each case?

13. **(Computing)** Modify `ray_sphere` to return the **surface normal** at the hit point, the unit vector $(\mathbf{p} - \mathbf{c})/r$, and test it on the worked example.

▼ **Answers**

1. Standard form $-3x^2 + 2x + 5 = 0$: $a = -3$, $b = 2$, $c = 5$.
   $\Delta = 4 - 4(-3)(5) = 4 + 60 = 64 > 0$: two real roots (and since $64 = 8^2$ they are rational: $x = (-2 \pm 8)/(-6)$, i.e. -1 and $5/3$).

2. (a) Vertex $(-2, 7)$, axis $x = -2$, maximum ($a = -3 < 0$). (b) Roots $4, -6$; axis $x = (4 + (-6))/2 = -1$; $y(-1) = (-5)(5) = -25$; vertex $(-1, -25)$, minimum.

3. (a) $(x + 5)(x - 4) = 0$: $x = -5, 4$. (b) $ac = -24 = -12 \cdot 2$, $-12 + 2 = -10$: $3x^2 - 12x + 2x - 8 = 3x(x - 4) + 2(x - 4) = (3x + 2)(x - 4)$: $x = -\frac{2}{3}, 4$. (c) $4x^2 - 9 = (2x - 3)(2x + 3) = 0$: $x = \pm\frac{3}{2}$.

4. (a) $x^2 - 8x = -3$; add 16: $(x - 4)^2 = 13$; $x = 4 \pm \sqrt{13}$. (b) Divide by 3: $x^2 + 4x = \frac{1}{3}$; add 4: $(x + 2)^2 = \frac{13}{3}$; $x = -2 \pm \sqrt{13/3} = -2 \pm \frac{\sqrt{39}}{3}$.

5. $\Delta = 9 + 56 = 65$; $x = \dfrac{-3 \pm \sqrt{65}}{4}$; $\sqrt{65} \approx 8.0623$; $x \approx 1.266$ or $x \approx -2.766$. Vieta check: sum $-1.500 = -3/2$ ✓, product $\approx -3.502 \approx -7/2$ ✓.

6. $y = -(x^2 - 6x) - 5 = -\big((x - 3)^2 - 9\big) - 5 = -(x - 3)^2 + 4$. Vertex $(3, 4)$, maximum. Roots: $(x - 3)^2 = 4$, $x = 1, 5$, so $y = -(x - 1)(x - 5)$. Sketch: cap through $(1, 0)$, $(5, 0)$, peak $(3, 4)$, y-intercept $(0, -5)$.

7. $2x^2 - 5x - 3 = (2x + 1)(x - 3)$, roots $-\frac{1}{2}, 3$, upward parabola, ≤ 0 between: $-\frac{1}{2} \le x \le 3$. Second: $\Delta = 16 - 20 < 0$ and $a > 0$, so it is always positive: all real x.

8. One root when $\Delta = k^2 - 36 = 0$: $k = \pm 6$ (giving $(x \pm 3)^2$). No real roots when $k^2 < 36$: $-6 < k < 6$.

9. $h(t) = 25 + 20t - 5t^2$. Hits water: $5t^2 - 20t - 25 = 0$, $t^2 - 4t - 5 = 0$, $(t - 5)(t + 1) = 0$, $t = 5$ s. Greatest height at $t = 20/10 = 2$ s: $25 + 40 - 20 = 45$ m above the sea.

10. Sides $s_1 + s_2 = 18$, $s_1 s_2 = 72$. They are the roots of $x^2 - 18x + 72 = 0 = (x - 6)(x - 12)$: 6 cm by 12 cm.

11. $\dfrac{1}{2}\left(\dfrac{-b + \sqrt{\Delta}}{2a} + \dfrac{-b - \sqrt{\Delta}}{2a}\right) = \dfrac{1}{2} \cdot \dfrac{-2b}{2a} = -\dfrac{b}{2a}$. The square roots cancel, which is the symmetry of the parabola in algebraic form.

12. For $b = 10^5$ the naive small root is $-1.0000003385 \times 10^{-5}$ against the true $-1.0000000001 \times 10^{-5}$: a relative error of about 3×10^{-7}, so only 6 or 7 correct digits (roughly 10 lost, consistent with error $\approx b^2 \times 10^{-16} = 10^{-6}$). For $b = -10^{12}$, $b^2 - 4 = 10^{24} - 4$ rounds to exactly 10^{24}, the square root is exactly 10^{12}, and the naive small root is 0.0: **every** digit lost. The stable version gives 10^{-12} correctly.
13. After computing `t`, form `p = [o + t * d for o, d in zip(origin, direction)]` and return `[(pi - ci) / radius for pi, ci in zip(p, centre)]`. For the worked example the normal is approximately $(-0.866, 0.5, 0)$, pointing back towards the camera, as it should.

8.18. Mastery check

1. Without solving, how many real roots does $4x^2 - 20x + 25 = 0$ have? What is special about this quadratic?
2. Solve $x^2 - 5x + 6 = 0$ and $x^2 - 5x + 5 = 0$. Which method did you use for each, and why?
3. Starting from $ax^2 + bx + c = 0$, derive the quadratic formula, justifying each step.
4. Write $y = 3x^2 + 12x + 7$ in vertex form. State its minimum value and where it occurs.
5. Solve $x^2 + 2x - 15 < 0$, giving your answer in interval notation.
6. A rectangular field is to be fenced on all four sides with 240 m of fencing. What dimensions maximise its area, and what is the maximum area?
7. The roots of $x^2 + 10^9 x + 1 = 0$ are needed in double-precision floating point. Which formula gives the root near 0 accurately, and why does the textbook formula fail?
8. Set up (but do not solve) the quadratic in t for a ray from $(0, 0, 0)$ in direction $(0, 0, 1)$ meeting a sphere of radius 1 centred at $(0, 0, 4)$. Then solve it and interpret both roots.

Decision rule: All eight correct without hesitation → move on to Chapter 9 (or Chapter 11, which builds directly on the forms and transformations here). Items 1–3 shaky → reread Section 8.5 and Section 8.6; the derivation is the heart of the chapter. Items 4–6 shaky → reread Section 8.8, Section 8.13 and Section 8.14. Items 7–8 shaky → reread Section 8.11 and Section 8.15. Stuck on more than three → work the chapter exercises first, then retry.

▼ **Mastery check answers**

1. $\Delta = 400 - 400 = 0$: exactly one (repeated) root. The quadratic is a perfect square, $(2x - 5)^2$, and its graph touches the axis at $x = \frac{5}{2}$.
2. $x^2 - 5x + 6 = (x - 2)(x - 3)$: $x = 2, 3$. Factoring, because $\Delta = 1$ is a perfect square and the factor pair is obvious. For $x^2 - 5x + 5$: $\Delta = 25 - 20 = 5$, not a square, so use the formula: $x = \frac{5 \pm \sqrt{5}}{2} \approx 3.618, 1.382$.

3. See Section 8.6: divide by a; move c/a across; add $(b/2a)^2$ to both sides; write the left side as $(x + b/2a)^2$ and the right side over $4a^2$ as $(b^2 - 4ac)/4a^2$; take $\pm$ square roots (requires $b^2 - 4ac \geq 0$); subtract $b/2a$.

4. $3(x^2 + 4x) + 7 = 3\big((x+2)^2 - 4\big) + 7 = 3(x+2)^2 - 5$. Minimum value -5 at $x = -2$.

5. $(x + 5)(x - 3) < 0$; upward parabola, negative between the roots: $-5 < x < 3$, i.e. $(-5, 3)$.

6. Sides x and $120 - x$; area $A = x(120 - x)$, roots $0, 120$, vertex at $x = 60$. A 60 m by 60 m square, area 3600 m^2.

7. Use $q = -\frac{1}{2}\left(b + \mathrm{sign}(b)\sqrt{\Delta}\right)$, then the small root is c/q. Here $q \approx -10^9$ and $c/q \approx -10^{-9}$. The textbook formula computes the small root as $(-b + \sqrt{\Delta})/2$ with $b = 10^9$, subtracting two numbers that agree in essentially all their digits (in fact $b^2 - 4$ rounds to b^2, so the naive result is exactly 0): catastrophic cancellation.

8. $\mathbf{f} = (0, 0, -4)$, $a = 1$, $b = 2(\mathbf{d} \cdot \mathbf{f}) = -8$, $c = 16 - 1 = 15$:
$t^2 - 8t + 15 = 0 = (t - 3)(t - 5)$. The ray enters the sphere at $t = 3$ (point $(0, 0, 3)$, the visible surface) and leaves at $t = 5$ (point $(0, 0, 5)$, the back of the sphere).

Part III: Geometry

Chapter 9. Euclidean Geometry: Shapes, Angles, Similarity, and Proof

A carpenter checks that a corner is square by measuring 3, 4 and 5 along the boards. A game engine decides whether a bullet hit a player by comparing two squared distances. A phone shrinks a 4000-pixel photo to fit a 1000-pixel screen without distorting anyone's face. All three are the same small body of ideas: angles, triangles, similarity, the Pythagorean theorem, circles, area and volume. This chapter rebuilds that toolkit, and it also reintroduces the habit geometry is famous for: proving things from a short list of starting rules.

Where this chapter sits

You need: arithmetic with fractions and square roots (Chapter 2, Chapter 3), solving linear equations (Chapter 5), and comfort rearranging formulas (Chapter 4). The difference of squares and other identities from Chapter 7 appear in a few proofs.

This chapter unlocks: coordinate geometry (Chapter 10), where every fact here becomes algebra; trigonometry (Chapter 13), which is the study of similar right triangles; vectors (Volume 2, Chapter 26), where the Pythagorean theorem becomes the length of a vector; and the habit of proof that Volume 3, Chapter 22 makes formal. Area-by-slicing is the first glimpse of integration (Volume 2, Chapter 19).

9.1. Points, lines, planes, and the rules of the game

9.1.1. The problem

Around 300 BCE, Euclid of Alexandria set out to do something unusual. Instead of collecting useful facts about shapes, which the Egyptians and Babylonians had been doing for over a thousand years, he wanted every fact to follow from a handful of statements so obvious that nobody could reasonably reject them. If you accept the starting rules, you must accept everything built from them.

That is exactly how a software engineer thinks about a type system or a set of database constraints: fix a small set of axioms, and every guarantee downstream is earned rather than assumed. Geometry was the first place humans did this systematically.

9.1.2. Vocabulary and notation

Three words are **undefined**: we describe them, but we don't define them in terms of anything simpler. Every other term is defined from them.



Point, line, plane

A **point** is a location with no size. Points are named with capital letters: A, B, P.

A **line** is straight, has no thickness, and extends forever in both directions. The line through A and B is written $\overleftrightarrow{AB}$.

A **plane** is a flat surface with no thickness that extends forever in every direction. A sheet of paper is a finite piece of one.

Points on the same line are **collinear**; points (or lines) in the same plane are **coplanar**.



Segment, ray, length, midpoint

A **line segment** $\overline{AB}$ is the part of a line between endpoints A and B. Its **length** is written AB (no bar). A **ray** $\overrightarrow{AB}$ starts at A and continues forever through B.

The **midpoint** of $\overline{AB}$ is the point M on it with $AM = MB$. Two segments with equal lengths are **congruent**, written $\overline{AB} \cong \overline{CD}$.



Axiom, postulate, theorem, lemma, corollary

An **axiom** or **postulate** is a statement accepted without proof (Euclid used "postulate" for geometric ones, "common notion" for general logic such as "things equal to the same thing are equal to each other").

A **theorem** is a statement proved from axioms and earlier theorems. A **lemma** is a small theorem proved mainly as a step towards a bigger one. A **corollary** is a result that follows almost immediately from a theorem.

A **conjecture** is a statement believed true but not yet proved.

Euclid's five postulates, in modern language:

1. A straight segment can be drawn between any two points.
2. A segment can be extended indefinitely into a line.
3. A circle can be drawn with any centre and any radius.
4. All right angles are equal.
5. **(The parallel postulate.)** Through a point not on a given line, there is exactly one line parallel to the given line.

The first four look like rules for using a ruler and compass. The fifth is different in character, and it is the one that powers most of this chapter: the angle sum of a triangle, similarity, and the Pythagorean theorem all depend on it.



The postulate that wouldn't go away

For two thousand years mathematicians tried to prove the parallel postulate from the other four and failed. In the 1820s Lobachevsky, Bolyai and Gauss realised why: you can replace it with "there are *many* parallels through the point" and get a perfectly consistent **hyperbolic geometry**, where triangle angles add to **less** than 180°. On the surface of a sphere, where "lines" are great circles, there are *no* parallels and triangle angles add to **more** than 180°. Draw a triangle from the North Pole down to the equator, a quarter of the way around, and back up: it has three right angles. Einstein's general relativity uses non-Euclidean geometry to describe curved spacetime, and GPS satellites must correct for it.

US curriculum

Undefined terms, postulates and the vocabulary of points, lines and planes open the high-school **Geometry** course (typically grade 9 or 10, after Algebra I). Common Core labels this strand **G-CO** (Congruence).

9.2. Angles and angle facts

9.2.1. The problem

Two roads cross. One makes a 63° angle with the other. What are the other three angles at the junction? Now a third road runs parallel to the first. What angles does it make with the second road? You should be able to answer both without a protractor, and the reasons are worth knowing because they are the first links in every geometric proof.

9.2.2. Vocabulary and notation



Angle and its measure

An **angle** is formed by two rays with a common endpoint, the **vertex**. The angle with vertex B and rays through A and C is written $\angle ABC$ (vertex in the middle) or just $\angle B$. Its **measure** is written $m\angle ABC$ in US textbooks, though most people just write $\angle ABC = 40°$.

Angles are measured in **degrees**: a full turn is $360°$. (In Chapter 13 you'll meet **radians**, where a full turn is 2π.)

Kinds of angle

- **Acute**: between $0°$ and $90°$.
- **Right**: exactly $90°$, marked with a small square in diagrams.
- **Obtuse**: between $90°$ and $180°$.
- **Straight**: exactly $180°$ (a straight line).
- **Reflex**: between $180°$ and $360°$.

Two lines that meet at a right angle are **perpendicular**, written $\ell_1 \perp \ell_2$. Lines in a plane that never meet are **parallel**, written $\ell_1 \parallel \ell_2$.

Angle pairs

- **Complementary** angles add to $90°$. **Supplementary** angles add to $180°$.
- **Adjacent** angles share a vertex and a side but don't overlap.
- A **linear pair** is two adjacent angles whose outer sides form a straight line; they are supplementary.
- **Vertical angles** (UK: **vertically opposite angles**) are the non-adjacent angles formed when two lines cross.

9.2.3. Intuition and why it works

Vertical angles are equal. When two lines cross, angles 1 and 2 form a straight line, so $\angle 1 + \angle 2 = 180°$. Angles 2 and 3 also form a straight line, so $\angle 2 + \angle 3 = 180°$. Both sums equal 180°, so $\angle 1 = 180° - \angle 2 = \angle 3$. That is the whole proof; we'll write it formally in Section 9.9.

Parallel lines cut by a transversal. A **transversal** is a line crossing two or more other lines. When it crosses two parallel lines it creates eight angles, and the parallel postulate forces the two crossings to look identical: the transversal meets ℓ_2 at exactly the same tilt as ℓ_1. So the four angles at the top crossing are copies of the four at the bottom.

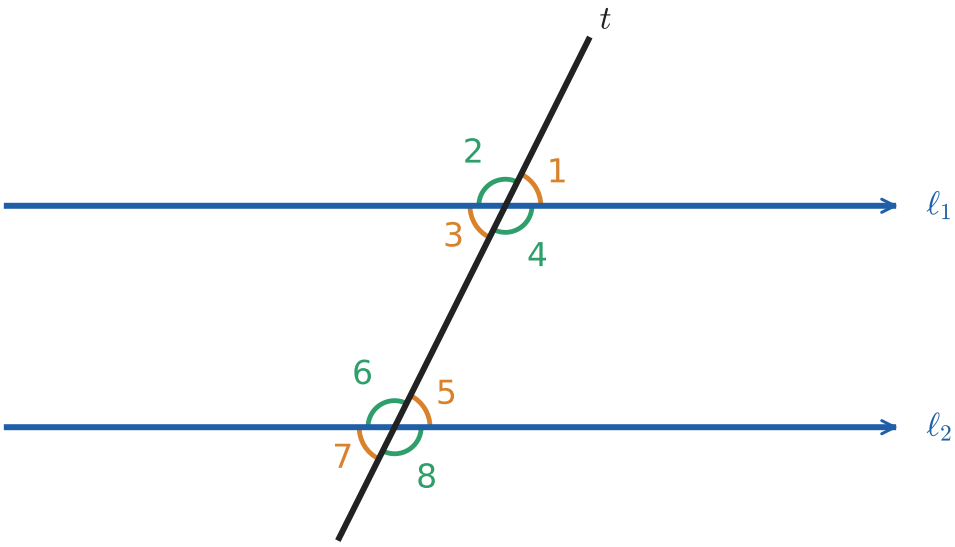


Figure 64. A transversal crossing two parallel lines. Only two different angle sizes appear.

The pairs have names, which US tests use heavily:

Name	Where they sit (using the figure)	Relationship
Corresponding	same position at each crossing: 1 & 5, 2 & 6, 3 & 7, 4 & 8	equal
Alternate interior	between the parallels, opposite sides of t: 3 & 5, 4 & 6	equal
Alternate exterior	outside the parallels, opposite sides: 1 & 7, 2 & 8	equal
Same-side (co-)interior	between the parallels, same side: 3 & 6, 4 & 5	supplementary

The table is really one fact: **every acute angle in the figure is equal, every obtuse angle is equal, and an acute plus an obtuse makes 180°.** (If the transversal is perpendicular, all eight are right angles.) The UK mnemonic is F-angles (corresponding), Z-angles (alternate) and C- or U-angles (co-interior), after the letter shapes they trace.

The facts also run backwards, and that is what makes them useful: if a transversal makes equal corresponding angles with two lines, those lines **are** parallel. This **converse** is how you prove lines parallel.

9.2.4. Worked example

Two parallel lines are cut by a transversal; angle 1 in the figure measures $63°$. Find all the others.

$$
\begin{aligned}
&\angle 3 = \angle 1 = 63^\circ && \text{(vertical angles)}\\
&\angle 2 = 180^\circ - 63^\circ = 117^\circ && \text{(linear pair with } \angle 1\text{)}\\
&\angle 4 = \angle 2 = 117^\circ && \text{(vertical angles)}\\
&\angle 5 = \angle 1 = 63^\circ && \text{(corresponding)}\\
&\angle 7 = 63^\circ, \quad \angle 6 = \angle 8 = 117^\circ && \text{(same reasoning at the lower crossing)}
\end{aligned}
$$

Check: $\angle 4 + \angle 5 = 117^\circ + 63^\circ = 180^\circ$, as same-side interior angles must.

A typical algebraic version: same-side interior angles measure $(3x + 15)^\circ$ and $(2x - 5)^\circ$. Since they are supplementary,

$$
\begin{aligned}
(3x + 15) + (2x - 5) &= 180\\
5x + 10 &= 180\\
5x &= 170\\
x &= 34
\end{aligned}
$$

so the angles are $3(34) + 15 = 117^\circ$ and $2(34) - 5 = 63^\circ$.



> *Looking parallel is not being parallel*
>
> The equal-angle facts hold **only** if the lines are parallel. If a problem doesn't state it (or mark the lines with matching arrowheads), you may not assume it, however parallel the drawing looks. Diagrams in geometry problems are routinely "not drawn to scale".

9.2.5. Where you'd use this

Navigation (bearings). A bearing is an angle measured clockwise from north. North lines at two different places are parallel, so the path between them is a transversal. If a boat sails from A to B on a bearing of 063°, the angle at B between north and the way back is the co-interior partner of 63°, namely 117° measured anticlockwise, so the return bearing is $360^\circ - 117^\circ = 243^\circ$. That's the rule every navigator and orienteer uses: **back bearing** = **bearing** $\pm 180^\circ$.

DIY. A picture frame's corners are 90°, so each of the two pieces meeting at a corner is cut at 45° (the cut bisects the corner). A hexagonal mirror frame has interior angles of 120°, so each cut is at 60° to the edge of the strip, which a mitre saw (whose dial reads 0° for a square cut) calls 30°.

Physics. Light reflects off a mirror at the same angle it arrives. Put two mirrors at a right angle and a ray that hits both comes back exactly parallel to the way it came, whatever angle it arrived at: the two reflections turn it through $2 \times 90^\circ = 180^\circ$. Bicycle reflectors, road "cat's eyes" and the reflector arrays the Apollo astronauts left on the Moon (used to measure the Earth–Moon distance by laser to within centimetres) are three-mirror versions of this corner.

Computing. In a 2D game, a ball bouncing off a vertical wall keeps its vertical velocity and flips the sign of its horizontal velocity, `vx = -vx`. Equal angles in and out is the geometric fact; one sign flip is its implementation. Ray tracers do the same thing for every reflective surface, using vectors (Volume 2, Chapter 26).

US curriculum

Angle pairs and parallel-line angle facts appear first in **grade 7–8 math** (Common Core 7.G.B.5, 8.G.A.5) and are revisited with proof in high-school **Geometry**.

9.3. Triangles: angle sum and congruence

9.3.1. The problem

Why do the angles of every triangle add to 180°? You've known the fact since childhood, but it's not obvious: a long thin triangle and a fat equilateral one look nothing alike. And a related engineering question: why are bridges and roof trusses built from triangles rather than rectangles? Both answers come from the same place.

9.3.2. Vocabulary



Classifying triangles

By sides:

- **Equilateral**: all three sides equal (so all angles $60°$).
- **Isosceles**: at least two sides equal. The equal sides are the **legs**; the third is the **base**; the angles opposite the legs are the **base angles**, and they are equal.
- **Scalene**: no two sides equal.

By angles: **acute** (all angles acute), **right** (one right angle), **obtuse** (one obtuse angle). In a right triangle the side opposite the right angle is the **hypotenuse** (always the longest side) and the other two are the **legs**.

Triangle ABC is written $\triangle ABC$.



Interior and exterior angles

The **interior angles** of a polygon are the angles inside it at each vertex. Extend one side past a vertex and the angle between the extension and the next side is an **exterior angle**. Each exterior angle forms a linear pair with its interior angle.

9.3.3. Why it works: the angle-sum theorem

Theorem. The interior angles of any triangle add to $180°$.

Proof. Take $\triangle ABC$ with angles α at A, β at B, γ at C. Through C, draw the line parallel to $\overline{AB}$ (the parallel postulate says there is exactly one).

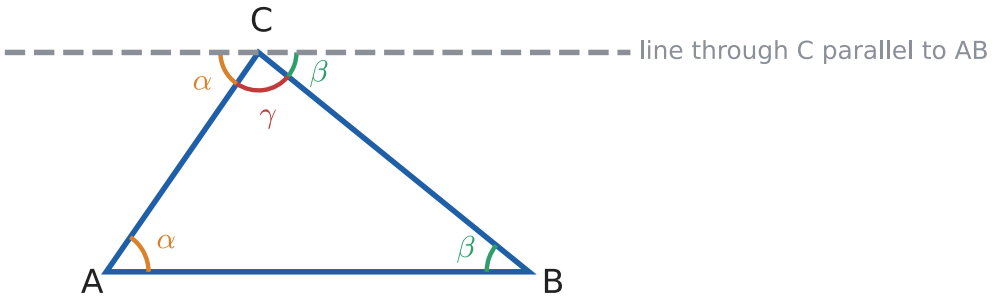


Figure 65. The parallel through C copies the base angles up to the top vertex.

Side $\overline{AC}$ is a transversal of the two parallel lines, so the angle between the new line and $\overline{CA}$ equals α (alternate interior angles). Likewise $\overline{BC}$ is a transversal, so the angle on the right at C equals β. Now the three angles at C, namely α, γ, β, sit side by side along a straight line, so

$$\alpha + \beta + \gamma = 180^\circ. \qquad \blacksquare$$

Notice what the proof *used*: the parallel postulate and the alternate-angle fact. On a sphere, where the parallel postulate fails, so does the theorem.

Corollary (exterior angle theorem). An exterior angle of a triangle equals the sum of the two interior angles not adjacent to it. If the exterior angle at C is δ, then $\delta = 180^\circ - \gamma = \alpha + \beta$.

Triangle inequality. Any side of a triangle is shorter than the sum of the other two: $AB < AC + CB$. "The straight path is the shortest." Lengths 2, 3 and 6 can't form a triangle because $2 + 3 < 6$. This inequality reappears as a defining property of every *distance function* in mathematics and computing.

9.3.4. Congruence

Two figures are **congruent** ($\cong$) if one can be moved onto the other exactly by sliding, rotating and possibly flipping it. Same shape, same size. For triangles, all six parts (three sides, three angles) match.

The practical question: **how little do you need to check** to know two triangles are congruent? Think of it as "how much information pins down a triangle uniquely". If you're told three side lengths, you can build only one triangle (up to moving it around): try it with three sticks. That is why a triangle frame is **rigid** while a rectangular frame can be pushed into a parallelogram. Four sides do not determine a quadrilateral; three sides do determine a triangle. Hence roof trusses.

Congruence criteria

Two triangles are congruent if they match in:

- **SSS**: all three sides.
- **SAS**: two sides and the angle **between** them (the **included** angle).
- **ASA**: two angles and the side between them.
- **AAS**: two angles and a non-included side (equivalent to ASA, since the third angle is then forced by the angle sum).
- **HL** (US) / **RHS** (UK): for right triangles, the hypotenuse and one leg.

In a congruence statement $\triangle ABC \cong \triangle DEF$ the **order matters**: A corresponds to D, B to E, C to F. US textbooks then use **CPCTC**, "corresponding parts of congruent triangles are congruent", to conclude that any other matching pair of parts is equal.

SSA and AAA don't work

AAA fixes the shape but not the size: every equilateral triangle has angles 60°, 60°, 60°. That gives **similarity**, not congruence.

SSA (two sides and a non-included angle) is ambiguous. Fix $\angle A = 30°$, $AB = 10$ and $BC = 6$. Swing the side of length 6 from B and it can hit the other ray from A at **two** different points, giving two non-congruent triangles. (This is the "ambiguous case" that resurfaces with the sine rule in Chapter 13.) The one exception is HL: when the given angle is a right angle, there's only one way to do it.

Try it: See both SSA triangles .

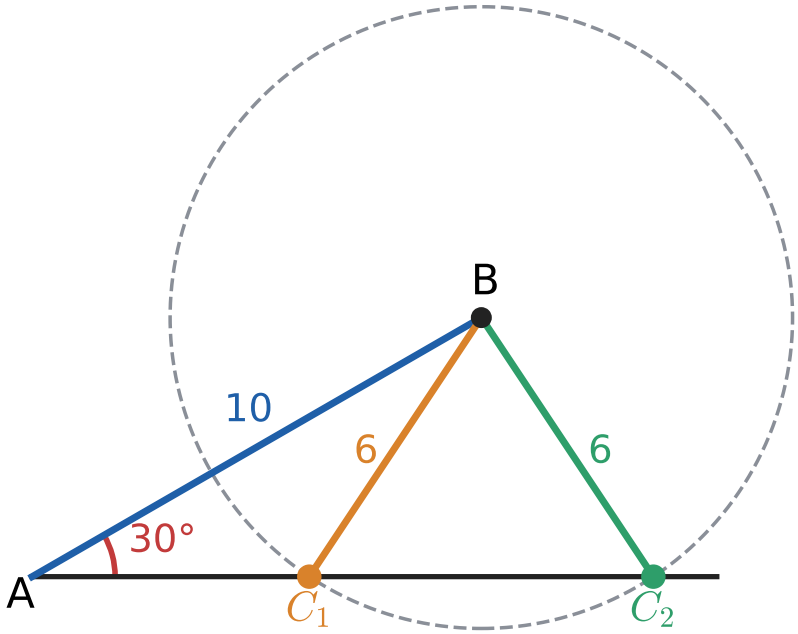


Figure 66. SSA is ambiguous: with $\angle A = 30°$, $AB = 10$ and $BC = 6$, the circle of radius 6 around B cuts the ray from A twice, giving two different triangles ABC_1 and ABC_2.

9.3.5. Worked example

In $\triangle PQR$ and $\triangle XYZ$: $PQ = XY = 7$, $\angle Q = \angle Y = 50°$, $QR = YZ = 9$. Are they congruent, and if so what is $\angle R$ given $\angle Z = 72°$?

1. The given angle at Q lies between sides PQ and QR; at Y it lies between XY and YZ. So this is SAS, and $\triangle PQR \cong \triangle XYZ$ with $P \leftrightarrow X$, $Q \leftrightarrow Y$, $R \leftrightarrow Z$.
2. By CPCTC, $\angle R = \angle Z = 72°$.
3. The angle sum then gives $\angle P = 180° - 50° - 72° = 58°$.

9.3.6. Where you'd use this

Home. A flat-pack bookcase that sways sideways is a rectangle, which can shear into a parallelogram without changing any side length. The thin board nailed across the back turns it into two triangles, and SSS says a triangle with fixed sides can't change shape. The same reason puts diagonal braces in garden gates (the brace runs from the bottom of the hinge side up to the top of the latch side, so it is pushed, not pulled) and triangles in every bicycle frame.

Engineering and surveying. Bridges and roof trusses (Warren and Pratt trusses are rows of triangles) and electricity pylons are triangulated for rigidity. Before satellites, national maps such as the UK Ordnance Survey were built from a network of triangles between hilltop "trig pillars": measure one baseline very accurately, measure angles, and the angle facts and similarity fix every other point.

Route planning and search. The triangle inequality gives you bounds for free. If a road trip is 30 km from A to B and 40 km from B to C, then A and C are at least $40 - 30 = 10$ km and at most 70 km apart in a straight line. Route planners such as A* use the straight-line distance as a lower bound on the road distance, which is what lets them skip hopeless routes safely. Nearest-neighbour indexes for

search use the same trick: if your query is 10 units from a reference point p, and an item is 3 units from p, that item is at least $10 - 3 = 7$ units from your query, so if you already have a match at distance 5 you never need to compute that item's distance at all.

Computer-science connection: rigidity and meshes

3D graphics models are **triangle meshes** for the same reason trusses are triangles: three points always determine a flat triangle (a plane), while four points generally don't lie in one plane, so a "quad" can be warped. GPUs rasterise triangles and nothing else; quads and polygons are split into triangles first. Physics engines model cloth and soft bodies as networks of springs arranged in triangles to keep them from collapsing.

US curriculum

The triangle angle sum appears in **grade 8** (8.G.A.5). Congruence criteria and their proofs are a central unit of high-school **Geometry** (Common Core G-CO.B.7–8, G-CO.C.10).

9.3.7. Exercises

1. Classify a triangle with angles $35°$, $55°$, $90°$ by angles, and say whether it can be isosceles.
2. An isosceles triangle has a vertex angle (between the legs) of $40°$. Find the base angles.
3. An exterior angle of a triangle is $130°$ and one of the non-adjacent interior angles is $48°$. Find the other two interior angles.
4. Which of these can be triangle side lengths: (a) 4, 5, 10 (b) 5, 5, 9 (c) 1, 2, 3?
5. Two triangles have $AB = DE$, $BC = EF$ and $\angle A = \angle D$. Must they be congruent? Explain.

▼ **Answers**

1. Right triangle (one angle is $90°$). It is scalene: an isosceles right triangle needs both acute angles equal ($45°$), and these are not.
2. The base angles are equal, so $2x + 40 = 180$, $2x = 140$, $x = 70°$.
3. Exterior angle theorem: $130 = 48 + y$, so $y = 82°$. The interior angle adjacent to the exterior angle is $180 - 130 = 50°$. Check: $48 + 82 + 50 = 180$.
4. (a) No: $4 + 5 = 9 < 10$. (b) Yes: $5 + 5 = 10 > 9$. (c) No: $1 + 2 = 3$ is not *greater* than 3; the "triangle" is flat (degenerate).
5. No. The angle at A is not between AB and BC, so this is SSA, which is ambiguous.

9.4. Similarity and scale factors

9.4.1. The problem

You want the height of a tree. You can't climb it, but it's a sunny day. You are 1.8 m tall and cast a 2.4 m shadow; the tree casts a 16 m shadow. How tall is the tree? Thales of Miletus supposedly measured the Great Pyramid this way around 600 BCE.

9.4.2. Intuition

Two figures are **similar** if one is a scaled copy of the other: same shape, possibly different size, possibly moved, rotated or flipped. Every length is multiplied by the same number k, the **scale factor**, and every angle is unchanged. A photo and its thumbnail are similar; a photo stretched to fill a different aspect ratio is not.

In the tree problem, the sun's rays are parallel, so the triangle formed by you, your shadow and the ray to the tip of your shadow has the same angles as the corresponding triangle for the tree. Same angles means same shape, so the ratios of sides are equal.



Similar figures and scale factor

$\triangle ABC \sim \triangle DEF$ ("is similar to") means corresponding angles are equal and corresponding sides are in a common ratio:

$$\frac{DE}{AB} = \frac{EF}{BC} = \frac{FD}{CA} = k,$$

where k is the **scale factor** from the first triangle to the second. $k > 1$ is an enlargement, $0 < k < 1$ a reduction. Congruence is the special case $k = 1$.

A **proportion** is an equation stating two ratios are equal, such as $\frac{a}{b} = \frac{c}{d}$. Its solution method, **cross-multiplying**, gives $ad = bc$.



Similarity criteria for triangles

- **AA**: two pairs of equal angles (the third pair then follows from the angle sum).
- **SSS~**: all three pairs of sides in the same ratio.
- **SAS~**: two pairs of sides in the same ratio, and the included angles equal.

AA is the workhorse: angles are usually easier to get than lengths, and parallel lines hand them to you for free. For instance, a line drawn across a triangle parallel to one side cuts off a smaller triangle similar to the original (the corresponding angles are equal). That fact is called the **side-splitter** or **basic proportionality theorem**.

9.4.3. The scaling laws

If lengths scale by k, what happens to area and volume?

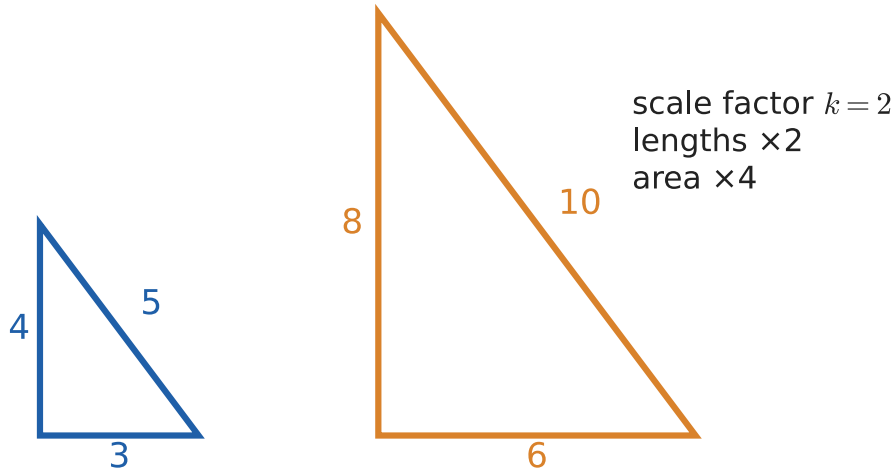


Figure 67. Scaling by k = 2 doubles every length and quadruples the area.

A square of side s has area s^2. Scale it by k and its area is $(ks)^2 = k^2 s^2$. Any shape can be approximated by tiny squares, each of which scales by k^2, so:

$$\text{lengths} \times k, \qquad \text{areas} \times k^2, \qquad \text{volumes} \times k^3.$$

The 3-4-5 triangle has area $\frac{1}{2} \cdot 3 \cdot 4 = 6$; the 6-8-10 triangle has area $\frac{1}{2} \cdot 6 \cdot 8 = 24 = 2^2 \times 6$.

9.4.4. The method

Example 55. Procedure — solving a similar-triangles problem

Input: a figure with two triangles you suspect are similar, some known lengths, and one unknown.

Steps:

1. Prove similarity, usually by AA (look for shared angles, vertical angles, right angles, parallel lines).
2. Write the correspondence in order: $\triangle ABC \sim \triangle DEF$. Matching letters tell you which sides correspond ($AB \leftrightarrow DE$, etc.). Corresponding sides are opposite equal angles.
3. Write one proportion containing the unknown and three knowns, keeping each ratio in the same order (small/large on both sides, or large/small on both).
4. Cross-multiply and solve the linear equation.
5. Sanity check: does the bigger triangle have the bigger side?

Output: the unknown length (and, if needed, the scale factor k).

9.4.5. Worked example: the tree

Both triangles have a right angle (person and tree are vertical) and the same angle of elevation of the sun. By AA they are similar. Matching "height ↔ height" and "shadow ↔ shadow":

$$\frac{\text{tree height}}{\text{your height}} = \frac{\text{tree shadow}}{\text{your shadow}}$$

$$\frac{h}{1.8} = \frac{16}{2.4}$$

$$2.4h = 1.8 \times 16 = 28.8 \qquad \text{(cross-multiply)}$$

$$h = \frac{28.8}{2.4} = 12 \text{ m}$$

Sanity check: the scale factor is $k = 16/2.4 = 6.\overline{6}$, and $1.8 \times 6.\overline{6} = 12$. The tree's shadow is much longer than yours, and so is the tree.

Alternative approach. Use the ratio *within* one triangle instead of *between* triangles: your height-to-shadow ratio is $1.8/2.4 = 0.75$, and the same ratio holds for the tree, so $h = 0.75 \times 16 = 12$. This "within" ratio is precisely what trigonometry will name the **tangent** of the sun's elevation angle (Chapter 13). Choosing a method: use "between" ratios when a scale factor is given or obvious (maps, models); use "within" ratios when you have a single triangle's shape and want to apply it to another, which is the trigonometric view.



Mixing up the correspondence

The most common error is matching the wrong sides. Always match by **angles**: the side opposite the right angle corresponds to the side opposite the right angle, and so on. Writing the similarity statement with vertices in matching order prevents the mistake. A second error is applying k to areas: a map at scale 1:50 000 shows a region whose real area is $50\,000^2 = 2.5 \times 10^9$ times larger, not 50 000 times.

9.4.6. Where you'd use this

- **Maps and plans:** on a 1:25 000 walking map, 1 cm stands for 25 000 cm = 250 m, so a 6.4 cm stretch of path is $6.4 \times 250 = 1600$ m. Areas scale by $25\,000^2$: a field covering 2 cm² of map is $2 \times 6.25 \times 10^8 \text{ cm}^2 = 125\,000 \text{ m}^2$, or 12.5 hectares. A 1:100 architect's drawing works the same way: 4.2 cm on paper is a 4.2 m wall.
- **Optics:** a pinhole camera projects a similar (inverted) image; the ratio of image height to object height equals the ratio of the image distance to the object distance. A 2 m tall person 20 m from a pinhole box 10 cm deep makes an image $2 \times 0.1/20 = 0.01$ m, i.e. 1 cm tall. Your eye and every camera lens obey the same ratio.
- **Cooking:** a round 20 cm cake recipe scaled to a 30 cm tin of the same depth needs $(30/20)^2 = 2.25$ times the ingredients, not 1.5 times, because the batter fills an area.

- **The square–cube law:** scale an animal up by k and its weight (volume) grows like k^3 while the strength of its bones (cross-sectional area) grows only like k^2. That's why giant ants in films are physically impossible and why elephants have thick legs.

Computer-science connection: image scaling and aspect ratio

Resizing an image **preserves similarity** only if both dimensions use the same scale factor. To fit a 4000 × 3000 photo into a 1000-pixel-wide slot:

```python
def scale_image_size(w, h, target_w):
    k = target_w / w                    # one scale factor for both axes
    return target_w, round(h * k), k * k   # new size and pixel-count ratio

print(scale_image_size(4000, 3000, 1000))
```

```
(1000, 750, 0.0625)
```

The height becomes 750, and the image now has $k^2 = 1/16$ of the pixels, which is why a 4× downscale cuts the file roughly sixteen-fold. Using different factors on the two axes ("stretch to fill") breaks similarity and distorts faces. CSS `object-fit: contain` and `cover` are both similarity-preserving: `contain` uses the smaller of the two possible scale factors so the image fits inside the box, `cover` uses the larger so it fills the box and crops the overflow.

Similarity also underlies **perspective projection** in 3D graphics: a point at depth z lands on the screen at $x_{\text{screen}} = f \cdot x/z$, a similar-triangles ratio with the focal distance f.



Shadow measurer

Drag the sun's elevation angle and the height of a pole. Watch the shadow length change, and see the "height / shadow" ratio stay constant for every object in the scene at a given sun angle. Then drag a second slider that scales a triangle by k and read off its perimeter ($\times k$) and area ($\times k^2$) live.

US curriculum

Scale drawings start in **grade 7** (7.G.A.1); similarity via dilations and the AA criterion are **grade 8** and high-school **Geometry** (G-SRT.A–B).

9.4.7. Exercises

1. $\triangle ABC \sim \triangle PQR$ with $AB = 4$, $BC = 6$, $PQ = 10$. Find QR and the scale factor.
2. A 1:25 scale model car is 18 cm long. How long is the real car in metres? If the model's paint covers 0.12 m², how much paint area does the real car have?

3. A 2 m post casts a 3.5 m shadow at the same time a building casts a 28 m shadow. How tall is the building?
4. Two similar cylinders have heights 5 cm and 15 cm. The small one holds 200 ml. How much does the large one hold?

▼ **Answers**

1. $k = PQ/AB = 10/4 = 2.5$, so $QR = 2.5 \times 6 = 15$.
2. Length: $18 \times 25 = 450$ cm $= 4.5$ m. Area scales by $25^2 = 625$: $0.12 \times 625 = 75$ m².
3. $h/2 = 28/3.5 = 8$, so $h = 16$ m.
4. $k = 15/5 = 3$, volume scales by $3^3 = 27$: $200 \times 27 = 5400$ ml $= 5.4$ litres.

9.5. The Pythagorean theorem

9.5.1. The problem

A 5 m ladder leans against a wall with its foot 1.4 m from the wall. How high up the wall does it reach? A television is sold as "55-inch" (the diagonal) with a 16:9 aspect ratio; how wide is it? A game character is at $(3, 4)$ relative to an enemy; how far away is that? All three questions are the same question.

9.5.2. The theorem



Pythagorean theorem

In a right triangle with legs a and b and hypotenuse c,

$$a^2 + b^2 = c^2.$$

Converse: if a triangle's sides satisfy $a^2 + b^2 = c^2$, the angle opposite c is a right angle.

A **Pythagorean triple** is a set of positive whole numbers satisfying the equation: (3, 4, 5), (5, 12, 13), (8, 15, 17), (7, 24, 25), and any multiple such as (6, 8, 10).

9.5.3. Why it works: a proof you can see

There are hundreds of proofs. This one is the most convincing at a glance.

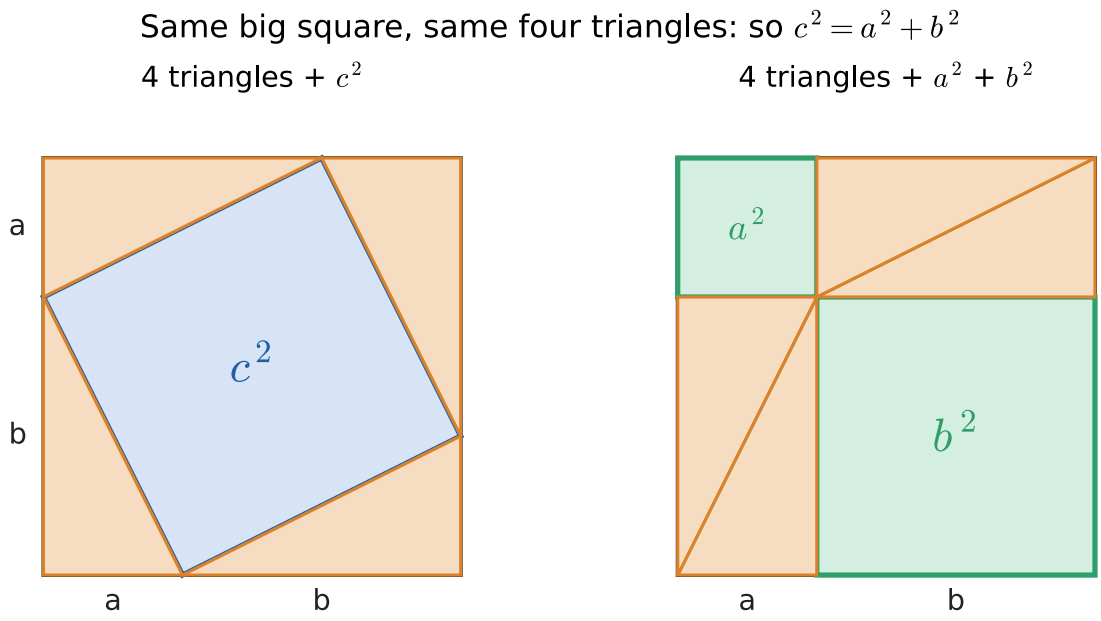


Figure 68. Two arrangements of the same square.

Take four copies of the right triangle with legs a, b and hypotenuse c. Arrange them inside a square of side $a + b$ in two ways.

- **Left:** the triangles sit in the corners, hypotenuses facing inwards. The hole in the middle is a square of side c. (It really is a square: at each corner of the hole, the two acute angles of the triangles plus the hole's angle form a straight line, and the two acute angles add to $90°$, so the hole's angle is $90°$.)
- **Right:** slide the triangles into two rectangles of size $a \times b$. The holes are now a square of side a and a square of side b.

The big square and the four triangles are identical in both pictures, so the uncovered areas must match: $c^2 = a^2 + b^2$. ■

The same picture gives an algebraic proof. Count the left-hand big square two ways:

$$\begin{aligned} (a+b)^2 &= c^2 + 4 \cdot \tfrac{1}{2}ab && \text{(the hole plus four triangles)} \\ a^2 + 2ab + b^2 &= c^2 + 2ab && \text{(expand the square)} \\ a^2 + b^2 &= c^2 && \text{(subtract } 2ab\text{)} \end{aligned}$$

Why the converse is true. Suppose $\triangle ABC$ has $a^2 + b^2 = c^2$. Build a genuine right triangle with legs a and b; by the theorem its hypotenuse is $\sqrt{a^2 + b^2} = c$. The two triangles now agree in all three sides, so they are congruent by SSS, and the original angle opposite c is a right angle too.

9.5.4. The method

Example 56. Procedure — finding a side of a right triangle

Input: a right triangle with two known sides.

Steps:

1. Identify the hypotenuse c: the side opposite the right angle, the longest side.
2. If you want the hypotenuse: $c = \sqrt{a^2 + b^2}$.
3. If you want a leg: $a = \sqrt{c^2 - b^2}$ (the hypotenuse squared minus the other leg squared).
4. Take the positive square root (lengths are positive). Leave an exact surd if asked, otherwise round at the end only.
5. Sanity check: the hypotenuse must be longer than each leg but shorter than their sum.

Output: the missing side.

9.5.5. Worked examples

The ladder. The ladder is the hypotenuse ($c = 5$); one leg is $b = 1.4$; we want the height a. *Try it:* Solve the ladder triangle .

$$a^2 = c^2 - b^2 = 5^2 - 1.4^2 = 25 - 1.96 = 23.04$$
$$a = \sqrt{23.04} = 4.8 \text{ m}$$

(This is the 7-24-25 triple scaled by 0.2: $1.4, 4.8, 5$.)

The television. With aspect ratio 16:9, write width $= 16t$ and height $= 9t$ for some t. The diagonal is 55:

$$(16t)^2 + (9t)^2 = 55^2$$
$$256t^2 + 81t^2 = 3025$$
$$337t^2 = 3025$$
$$t = \frac{55}{\sqrt{337}} \approx \frac{55}{18.358} \approx 2.996$$

so the width is $16t \approx 47.9$ inches and the height $9t \approx 27.0$ inches.

The converse at work. Is a triangle with sides 9, 40, 41 a right triangle? $9^2 + 40^2 = 81 + 1600 = 1681 = 41^2$. Yes. And 6, 7, 9? $36 + 49 = 85 \neq 81 = 9^2$, so it isn't a right triangle. In fact if $a^2 + b^2 > c^2$ (with c the longest side) the triangle is **acute**, and if $a^2 + b^2 < c^2$ it is **obtuse**. The law of cosines in Chapter 13 makes this precise.

9.5.6. Distance in two and three dimensions

Two points (x_1, y_1) and (x_2, y_2) are the ends of the hypotenuse of a right triangle whose legs are the horizontal and vertical separations $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$. (Δ, Greek capital delta, means "change in".) So the distance is

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2}.$$

From $(1, 2)$ to $(4, 6)$: $\sqrt{3^2 + 4^2} = 5$. Chapter 10 develops this fully as the **distance formula**.

In 3D, apply the theorem twice. The diagonal of a box measuring $3 \times 4 \times 12$: the floor diagonal is $\sqrt{3^2 + 4^2} = 5$, and that floor diagonal together with the height forms another right triangle, so the space diagonal is $\sqrt{5^2 + 12^2} = 13$. In general $d = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$, and the pattern continues to any number of dimensions, which is how machine learning measures the distance between 768-dimensional embedding vectors (Volume 2, Chapter 26).



Forgetting which side is the hypotenuse

$a^2 + b^2 = c^2$ only applies to **right** triangles, and c must be the hypotenuse. Given legs 5 and 13, the hypotenuse is $\sqrt{194} \approx 13.93$, not 12; given a leg 5 and hypotenuse 13, the other leg is 12. Also: $\sqrt{a^2 + b^2} \neq a + b$. The square root does not distribute over addition: $\sqrt{9 + 16} = 5$, but $\sqrt{9} + \sqrt{16} = 7$.

9.5.7. Special right triangles

Two right triangles turn up so often that it pays to know their side ratios by heart. Both come from cutting a regular shape in half.

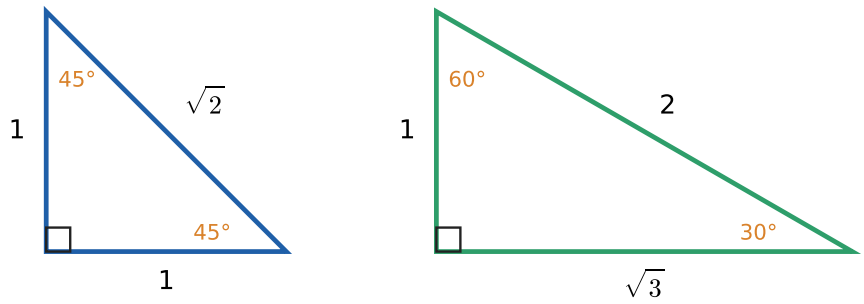


Figure 69. The two special right triangles.

45-45-90. Cut a unit square along its diagonal. Each half is an isosceles right triangle with legs 1 and hypotenuse $\sqrt{1^2 + 1^2} = \sqrt{2}$. Sides are in the ratio $1 : 1 : \sqrt{2}$.

30-60-90. Cut an equilateral triangle of side 2 down its line of symmetry. Each half has hypotenuse 2, short leg 1 (half the base), and angles $30°, 60°, 90°$. The long leg is $\sqrt{2^2 - 1^2} = \sqrt{3}$. Sides are in the ratio $1 : \sqrt{3} : 2$, with the short leg opposite the $30°$ angle.

Example. An equilateral triangle has side 10. Its height is the long leg of a 30-60-90 triangle with hypotenuse 10, short leg 5, so the height is $5\sqrt{3} \approx 8.66$ and the area is $\frac{1}{2} \cdot 10 \cdot 5\sqrt{3} = 25\sqrt{3} \approx 43.30$. In general, an equilateral triangle of side s has area $\frac{\sqrt{3}}{4}s^2$.

These ratios are exactly the values of sine, cosine and tangent at $30°, 45°, 60°$ that Chapter 13 will ask you to know: for example $\sin 30° = \frac{1}{2}$ is "short leg over hypotenuse".

9.5.8. Where you'd use this

- **Building:** to set out a square corner for a shed base, measure 0.9 m along one edge and 1.2 m along the other; the corner is square exactly when the diagonal is 1.5 m, because $0.9^2 + 1.2^2 = 0.81 + 1.44 = 2.25 = 1.5^2$. (This is 3-4-5 scaled by 0.3; builders use 6-8-10 feet for bigger corners.)
- **Shopping:** will a 65-inch 16:9 television fit a 1.5 m wide alcove? As in the worked example, $t = 65/\sqrt{337} \approx 3.541$ inches, so the screen is $16t \approx 56.7$ inches ≈ 144 cm wide (plus the bezel) and $9t \approx 31.9$ inches ≈ 81 cm tall. It fits, with about 6 cm to spare.
- **Mechanics:** two forces of 30 N and 40 N pulling at right angles have a resultant of $\sqrt{30^2 + 40^2} = 50$ N. A plane flying at 200 km/h due north through a 50 km/h wind from the west actually moves over the ground at $\sqrt{200^2 + 50^2} = \sqrt{42\,500} \approx 206.2$ km/h. Velocities, forces and displacements at right angles always combine this way, which is the doorway to vectors (Volume 2, Chapter 26).



Over 3,700 years old

The theorem is named after Pythagoras (c. 570–495 BCE), but the Babylonian clay tablet **Plimpton 322** (c. 1800 BCE) lists fifteen Pythagorean triples, including (12 709, 13 500, 18 541), which nobody finds by accident. Euclid's proof is Proposition 47 of Book I. US President James Garfield published an original proof in 1876 using a trapezoid made of three right triangles. Every primitive triple can be generated as $(m^2 - n^2,\ 2mn,\ m^2 + n^2)$; try $m = 2, n = 1$.

Computer-science connection: distance without square roots

Distance comparisons are everywhere in software: nearest-neighbour search, "is the player within 10 m of the trigger?", k-means clustering. Square roots are relatively expensive, and you rarely need them: because both sides are non-negative, $d \le r \iff d^2 \le r^2$. So compare **squared** distances.

```python
def circles_collide(c1, r1, c2, r2):
    dx = c2[0] - c1[0]
    dy = c2[1] - c1[1]
    return dx * dx + dy * dy <= (r1 + r2) ** 2   # no square root needed

def boxes_collide(a, b):
    """a, b = (xmin, ymin, xmax, ymax), axis-aligned boxes."""
    return a[0] <= b[2] and b[0] <= a[2] and a[1] <= b[3] and b[1] <= a[3]

print(circles_collide((0, 0), 3, (4, 5), 3))   # 41 > 36
print(circles_collide((0, 0), 3, (4, 4), 3))   # 32 <= 36
print(boxes_collide((0, 0, 2, 2), (1, 1, 3, 3)), boxes_collide((0, 0, 2, 2), (3, 0, 4,
1)))
```

```
False
True
True False
```

Two circles touch exactly when the distance between centres is at most the sum of the radii; that is the whole of **circle–circle collision detection**. Game engines usually test cheap bounding circles or axis-aligned bounding boxes first and run an exact polygon test only on the few pairs that pass. When you do need the actual distance, Python's `math.hypot(dx, dy)` and `math.dist(p, q)` compute it while avoiding overflow for huge values.



Pythagoras jigsaw

Drag the corner of a right triangle to change a and b. The app animates the four triangles sliding between the two arrangements, and displays a^2, b^2 and c^2 as numbers that always balance. Then unlock the right angle: as the angle opens past 90°, watch c^2 exceed $a^2 + b^2$.

Try it: Solve any triangle

US curriculum

The Pythagorean theorem and its converse are **grade 8** (Common Core 8.G.B.6–8), including distance between points. Special right triangles are taught in high-school **Geometry**, just before right-triangle trigonometry (G-SRT.C).

9.5.9. Exercises

1. Find the hypotenuse of a right triangle with legs 8 and 15.
2. A rectangle is 20 cm by 21 cm. How long is its diagonal?
3. Is a triangle with sides 10, 24, 26 right, acute or obtuse? What about 5, 6, 8?
4. A 45-45-90 triangle has hypotenuse 12. Find its legs, exactly.
5. A 30-60-90 triangle has long leg 9. Find the short leg and the hypotenuse.
6. A drone flies 30 m east, 40 m north and climbs 120 m. How far is it from where it started?

▼ **Answers**

1. $\sqrt{64 + 225} = \sqrt{289} = 17$.
2. $\sqrt{400 + 441} = \sqrt{841} = 29$ cm.
3. $100 + 576 = 676 = 26^2$: right. $25 + 36 = 61 < 64 = 8^2$: obtuse.
4. Legs ℓ with $\ell\sqrt{2} = 12$, so $\ell = 12/\sqrt{2} = 6\sqrt{2} \approx 8.49$.
5. Long leg $= \sqrt{3}\times$ short leg, so short leg $= 9/\sqrt{3} = 3\sqrt{3} \approx 5.20$; hypotenuse $= 2 \times 3\sqrt{3} = 6\sqrt{3} \approx 10.39$.
6. Ground distance $\sqrt{30^2 + 40^2} = 50$; total $\sqrt{50^2 + 120^2} = \sqrt{16\,900} = 130$ m.

9.6. Polygons, perimeter and area

9.6.1. The problem

A surveyor walks around a plot of land and records the corners as coordinates. What is its area? A tiler wants to know which regular shapes can tile a floor with no gaps. Both need the same two tools: angle sums of polygons and a reliable way to compute area.



Polygon vocabulary

A **polygon** is a closed plane figure made of straight segments (**sides** or **edges**) meeting at **vertices**. It is **simple** if its sides don't cross each other, and **convex** if every interior angle is less than $180°$ (equivalently, the segment joining any two interior points stays inside). A polygon with a "dent" is **concave**. A **diagonal** joins two non-adjacent vertices.

A **regular** polygon has all sides equal **and** all angles equal. Names by number of sides: triangle (3), quadrilateral (4), pentagon (5), hexagon (6), heptagon (7), octagon (8), decagon (10), n-gon in general.

Perimeter is the total length of the boundary. **Area** is the amount of surface enclosed, measured in square units.

Quadrilaterals

- **Parallelogram**: both pairs of opposite sides parallel (and therefore equal).
- **Rectangle**: a parallelogram with four right angles.
- **Rhombus**: a parallelogram with four equal sides.
- **Square**: both a rectangle and a rhombus.
- **Trapezoid** (US) / **trapezium** (UK): exactly one pair of parallel sides (some US texts say "at least one"). The parallel sides are the **bases**.
- **Kite**: two pairs of adjacent equal sides.

Watch the transatlantic trap: in the UK a **trapezium** has one pair of parallel sides; in the US a **trapezium** traditionally has *none*.

9.6.2. Angle sums of polygons

Pick one vertex of a convex n-gon and draw every diagonal from it. You cut the polygon into $n - 2$ triangles (each side except the two touching your vertex becomes the base of one triangle). Each triangle contributes $180°$, so

$$\text{sum of interior angles} = (n - 2) \times 180°.$$

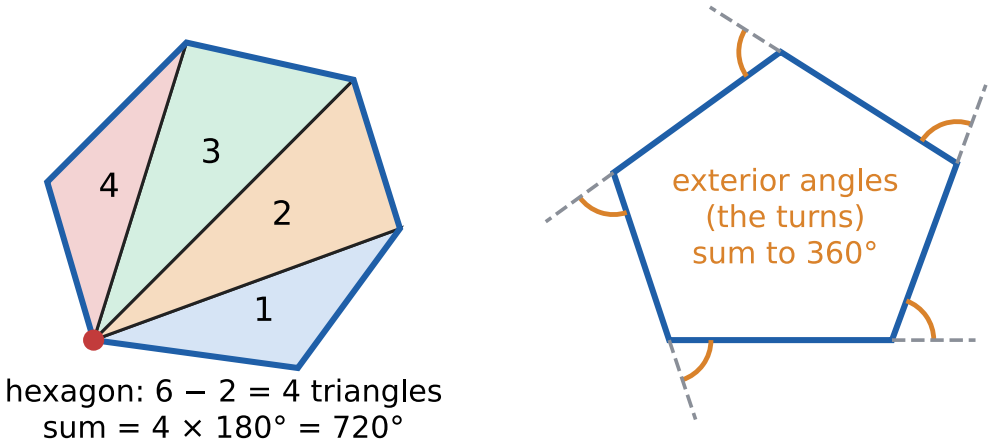


Figure 70. Left: fanning from one vertex cuts a hexagon into $6 - 2 = 4$ triangles. Right: walking round a polygon, the exterior angles are the turns you make, and one lap is one full turn.

For a regular polygon each interior angle is $\dfrac{(n - 2) \times 180°}{n}$: 60° (triangle), 90° (square), 108° (pentagon), 120° (hexagon), 135° (octagon).

The **exterior** angles are even simpler. Walk around any convex polygon: at each corner you turn through the exterior angle, and after one lap you have turned through exactly one full turn. So the exterior angles of **every** convex polygon add to $360°$, regardless of n. For a regular n-gon each

exterior angle is $360°/n$, which is the quickest way to find n: a regular polygon with interior angles of $156°$ has exterior angles of $24°$, so $n = 360/24 = 15$.



Only three regular tilings

For regular polygons to tile a floor, a whole number of interior angles must meet at each point to make $360°$. Six triangles (6×60), four squares (4×90) or three hexagons (3×120) work. Pentagons (108°) don't divide 360, and anything with more than six sides has angles over 120°, so fewer than three fit. That's why bees, bathroom tiles and graphics engines use exactly these three. Hexagonal grids are popular in strategy games because every neighbour of a hexagon is the same distance away, which isn't true of a square's diagonal neighbours.

9.6.3. Area formulas, derived

Every basic area formula comes from the rectangle by cutting and rearranging, which is worth seeing because it means you never need to memorise them blindly.

Rectangle: $A = bh$ (base b, height h). That's the definition of area in square units: b columns of h unit squares.

Parallelogram: cut the triangle off one end and slide it to the other end. You get a rectangle with the same base and the same **perpendicular** height: $A = bh$.

Triangle: two copies of any triangle fit together into a parallelogram with the same base and height. So

$$A_{\triangle} = \tfrac{1}{2}bh,$$

where h is the perpendicular height measured to the line containing the base (for an obtuse triangle that may fall outside the triangle).

Trapezoid with parallel sides a and b, height h: two copies, one rotated, form a parallelogram of base $a + b$, so $A = \tfrac{1}{2}(a + b)h$. ("Average of the parallel sides, times the height.")



Using the slant side as the height

The height in every area formula is **perpendicular** to the base. A parallelogram with sides 10 and 6 does **not** have area 60 unless it's a rectangle; tilt it and the area shrinks towards zero even though the side lengths don't change. Also keep units straight: lengths in cm give areas in cm² (and $1\ \text{m}^2 = 10\,000\ \text{cm}^2$, not 100).

9.6.4. Area from coordinates: the shoelace formula

The surveyor's problem: you have the corners $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ of a simple polygon, listed in order around the boundary. There's no obvious base or height. The **shoelace formula** (or **surveyor's formula**) handles any simple polygon, convex or not:

$$A = \frac{1}{2}\left|\sum_{i=1}^{n} (x_i\, y_{i+1} - x_{i+1}\, y_i)\right|, \qquad \text{with } (x_{n+1}, y_{n+1}) = (x_1, y_1).$$

Read the notation piece by piece: $\sum_{i=1}^{n}$ (capital sigma) means "add up the following expression for $i = 1, 2, \ldots, n$" (it's a `for` loop, and Chapter 14 covers it in detail); $|\cdot|$ is absolute value; and the condition at the end says "wrap around" so the last vertex pairs with the first.

Why it works (sketch). Each term $\frac{1}{2}(x_i y_{i+1} - x_{i+1} y_i)$ is the **signed** area of the triangle formed by the origin and the edge from vertex i to vertex $i+1$: positive if the edge goes anticlockwise around the origin, negative if clockwise. Adding these triangles for every edge, the parts outside the polygon cancel (they are swept once each way) and only the polygon's area remains. The sign of the total tells you the vertex order: positive for anticlockwise, negative for clockwise. (In Volume 2, Chapter 27 each term turns out to be a $\mathbf{2 \times 2}$ **determinant**.) The name comes from the criss-cross pattern of multiplications when you write the coordinates in two columns, like lacing a shoe.

Example 57. Procedure — shoelace formula

Input: vertices $(x_1, y_1), \ldots, (x_n, y_n)$ of a simple polygon in order (either direction), $n \geq 3$.

Steps:

1. List the vertices in a column and repeat the first one at the bottom.
2. For each consecutive pair of rows, compute the cross product $x_i y_{i+1} - x_{i+1} y_i$.
3. Add all the cross products.
4. Take the absolute value and halve it.

Output: the area of the polygon. (The sign before step 4 gives the orientation: positive = anticlockwise.)

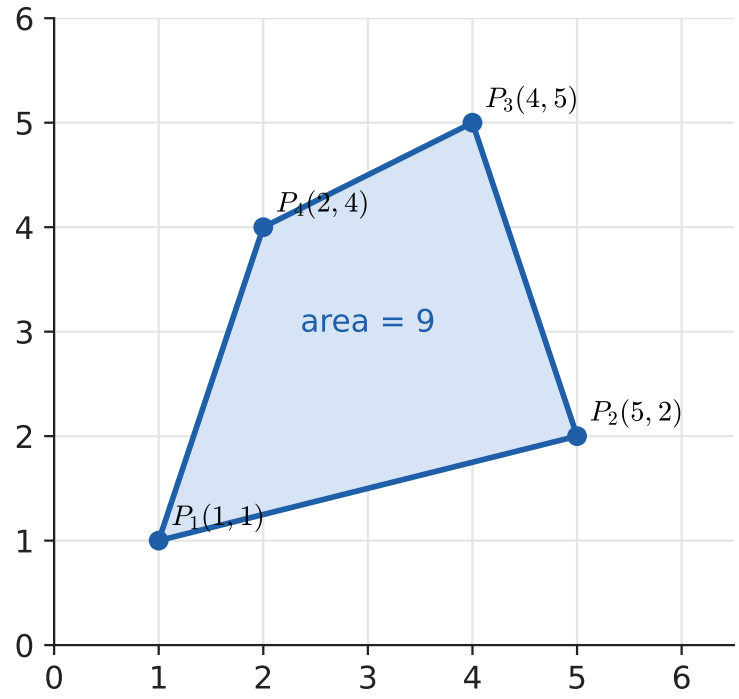


Figure 71. A quadrilateral given by its corners.

Worked example. Vertices $P_1(1,1)$, $P_2(5,2)$, $P_3(4,5)$, $P_4(2,4)$. *Try it:* Shoelace area calculator .

$$
\begin{aligned}
x_1y_2 - x_2y_1 &= 1\cdot 2 - 5\cdot 1 = -3 \\
x_2y_3 - x_3y_2 &= 5\cdot 5 - 4\cdot 2 = 17 \\
x_3y_4 - x_4y_3 &= 4\cdot 4 - 2\cdot 5 = 6 \\
x_4y_1 - x_1y_4 &= 2\cdot 1 - 1\cdot 4 = -2 \qquad \text{(wrap around)} \\
\text{sum} &= -3 + 17 + 6 - 2 = 18 \\
A &= \tfrac{1}{2}|18| = 9
\end{aligned}
$$

The sum is positive, so the vertices were listed anticlockwise.

Alternative approach: box and subtract. Draw the bounding rectangle from $x = 1$ to 5 and $y = 1$ to 5: area 16. Subtract the pieces of the box outside the polygon: below edge P_1P_2, a right triangle $\tfrac{1}{2}\cdot 4\cdot 1 = 2$; right of edge P_2P_3, $\tfrac{1}{2}\cdot 1\cdot 3 = 1.5$; above edge P_3P_4, $\tfrac{1}{2}\cdot 2\cdot 1 = 1$; left of edge P_4P_1, $\tfrac{1}{2}\cdot 1\cdot 3 = 1.5$; and one piece people forget, the 1×1 square in the top-left corner between $P_4(2,4)$ and the box corner $(1,5)$, which belongs to neither neighbouring triangle. Total removed: $2 + 1.5 + 1 + 1.5 + 1 = 7$, so $A = 16 - 7 = 9$. The box method works for simple shapes on grid paper, but that forgotten corner square shows how easy it is to get wrong; the shoelace formula is mechanical and never needs a picture.

Choosing a method. For triangles and standard quadrilaterals with known base and height, use the formulas. For any polygon given by coordinates, use the shoelace formula; it's exact, handles concave shapes, and is a five-line program. For lattice polygons (all vertices on integer points) there's also **Pick's theorem**: $A = I + B/2 - 1$, with I interior lattice points and B boundary lattice points. Here $I = 8$ and $B = 4$ (the edges pass through no other lattice points), so $A = 8 + 2 - 1 = 9$. Consistent.

```python
def shoelace_area(points):
    """Area of a simple polygon whose vertices are listed in order."""
    n = len(points)
    if n < 3:
        raise ValueError("a polygon needs at least 3 vertices")
    twice_area = 0
    for i in range(n):
        x_i, y_i = points[i]
        x_next, y_next = points[(i + 1) % n]   # wrap around to the start
        twice_area += x_i * y_next - x_next * y_i
    return abs(twice_area) / 2

print(shoelace_area([(1, 1), (5, 2), (4, 5), (2, 4)]))
print(shoelace_area([(0, 0), (4, 0), (4, 3)]))           # right triangle 4 x 3
print(shoelace_area([(0, 0), (0, 3), (4, 3), (4, 0)]))   # rectangle, clockwise
```

```
9.0
6.0
12.0
```



Self-intersecting polygons

The vertices must be in boundary order and the polygon must be **simple**. Feed it a "bow-tie" (vertices in the wrong order so edges cross) and the two lobes have opposite orientation and partly cancel, giving a meaningless number. Real GIS data occasionally contains such invalid polygons, which is why libraries like Shapely have an `is_valid` check. For very large coordinates (latitude/longitude converted to metres), subtract the first vertex from all the others before applying the formula to reduce floating-point cancellation error (Volume 2, Chapter 32).

9.6.5. Where you'd use this

- **Decorating:** a wall 4 m wide and 2.5 m high with a 0.8 m by 2.0 m door has $10 - 1.6 = 8.4$ m² to paint. Two coats of a paint that covers 12 m² per litre need $2 \times 8.4/12 = 1.4$ litres. For flooring, split an L-shaped room into rectangles (5×4 plus $3 \times 2 = 26$ m²) and add about 10% for cutting waste: buy 28.6 m².
- **Surveying and science:** land registries, farm subsidy claims and ecologists measuring habitat all record a boundary as GPS points and apply the shoelace formula. Hexagonal grids appear in nature (honeycomb, graphene) because hexagons tile with the least boundary for a given area.
- **Computing:** triangulating a polygon (the fan of triangles above, for a convex shape) is how a GPU draws it, and the shoelace formula is the area function in GIS libraries.

Computer-science connection: computational geometry

The cross product $x_1y_2 - x_2y_1$ inside the shoelace formula is the most important primitive in computational geometry. Its **sign** tells you whether three points turn left, turn right or are collinear, and from that one test you can build convex-hull algorithms (Graham scan), segment-intersection tests, and the winding direction of a font glyph's outline (TrueType outer contours run one way, holes the other). Mapping software computes the area of a land parcel, a lake or a delivery zone with exactly the function above.

US curriculum

Area of triangles, parallelograms and trapezoids is **grade 6** (6.G.A.1); polygon angle sums appear in **grade 8** or **Geometry**. Area from coordinates is part of **Geometry** (G-GPE.B.7), though the shoelace formula itself is usually met in math-competition training rather than the standard course.

9.6.6. Exercises

1. Find the sum of the interior angles of a decagon, and each interior angle if it is regular.
2. Each exterior angle of a regular polygon is $18°$. How many sides does it have?
3. A trapezoid has parallel sides 7 and 13 and height 4. Find its area.
4. Use the shoelace formula to find the area of the triangle with vertices $(0,0)$, $(6,1)$, $(2,5)$.
5. Explain why a parallelogram with sides 8 and 5 can have any area between 0 and 40.

▼ **Answers**

1. $(10-2)\times 180 = 1440°$; each angle $1440/10 = 144°$.
2. $n = 360/18 = 20$.
3. $\tfrac{1}{2}(7+13)\times 4 = 40$.
4. Terms: $0\cdot 1 - 6\cdot 0 = 0$; $6\cdot 5 - 2\cdot 1 = 28$; $2\cdot 0 - 0\cdot 5 = 0$. Sum 28, area 14.
5. Area is base times **perpendicular** height. With base 8, the height is $5\times$ (something between 0 and 1, depending on the tilt), so the area ranges from nearly 0 (flattened) up to 40 (a rectangle).

9.7. Circles

9.7.1. The problem

Is one 18-inch pizza more food than two 12-inch pizzas? (Most people guess no.) How long is the crust on a 45° slice? And a puzzle that has a lovely answer: if you stand anywhere on the edge of a circular stage and look at two fixed points on the edge, does the angle you see depend on where you stand?

9.7.2. Vocabulary



Parts of a circle

A **circle** is the set of all points in a plane at a fixed distance, the **radius** r, from a fixed point, the **centre**.

- **Diameter** d: a chord through the centre; $d = 2r$.
- **Circumference** C: the perimeter of the circle.
- **Chord**: a segment joining two points on the circle.
- **Arc**: a piece of the circle between two points. A **minor arc** is less than half the circle, a **major arc** more, a **semicircle** exactly half.
- **Sector**: the "pizza slice" region between two radii and an arc.
- **Segment** (of a circle): the region between a chord and its arc.
- **Tangent**: a line touching the circle at exactly one point, the **point of tangency**.
- **Secant**: a line crossing the circle at two points.
- **Central angle**: an angle with its vertex at the centre. **Inscribed angle**: an angle with its vertex on the circle, its sides being chords.



Pi

π (Greek lowercase pi) is the ratio of any circle's circumference to its diameter:

$$\pi = \frac{C}{d} = 3.14159\,26535\ldots$$

It is the **same for every circle** because all circles are similar: scaling a circle by k multiplies both C and d by k, leaving the ratio unchanged. π is **irrational** (its decimal expansion never repeats; Chapter 1) and even **transcendental** (it is not a root of any polynomial with integer coefficients).

9.7.3. Circumference, arc length, sector area

From the definition of π:

$$C = \pi d = 2\pi r.$$

For the area, cut the circle into many thin sectors and rearrange them alternately point-up and point-down.

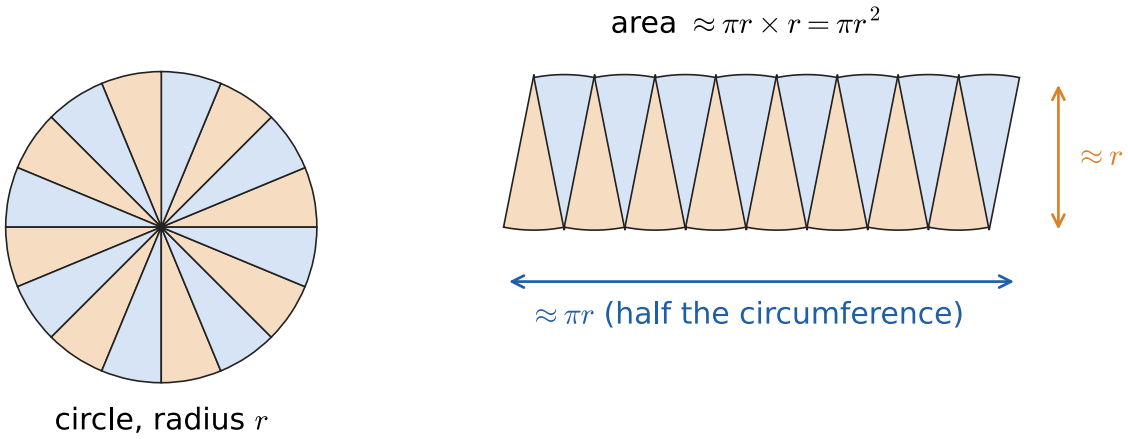


Figure 72. Slicing a circle and rearranging the slices.

The result is almost a parallelogram. Its top and bottom edges are each half the circumference, πr, and its height is almost exactly r. As the slices get thinner, the scalloped edges flatten and the shape becomes a true rectangle πr by r. So

$$A = \pi r \times r = \pi r^2.$$

"As the slices get thinner, the approximation becomes exact" is the idea of a **limit** (Volume 2, Chapter 16), and cutting a region into thin pieces and adding them up is exactly what an **integral** does (Volume 2, Chapter 19). You've just done calculus informally, as Archimedes did.

A sector or arc with central angle θ (in degrees) is the fraction $\theta/360$ of the whole:

$$\text{arc length} = \frac{\theta}{360} \cdot 2\pi r, \qquad \text{sector area} = \frac{\theta}{360} \cdot \pi r^2.$$

9.7.4. Worked examples

The pizza. Diameters 18 and 12 mean radii 9 and 6.

$$\begin{aligned} \text{one 18-inch}: &\quad \pi \cdot 9^2 = 81\pi \approx 254.5 \text{ in}^2 \\ \text{two 12-inch}: &\quad 2 \cdot \pi \cdot 6^2 = 72\pi \approx 226.2 \text{ in}^2 \end{aligned}$$

The single large pizza wins by about 12%. Area scales as the **square** of the diameter (scaling law again), and intuition consistently underestimates squares.

The slice. A 40 cm pizza ($r = 20$) cut into eight 45° slices.

$$\begin{aligned} \text{crust (arc)} &= \frac{45}{360} \cdot 2\pi \cdot 20 = \frac{1}{8} \cdot 40\pi = 5\pi \approx 15.71 \text{ cm} \\ \text{slice area} &= \frac{45}{360} \cdot \pi \cdot 20^2 = \frac{1}{8} \cdot 400\pi = 50\pi \approx 157.08 \text{ cm}^2 \end{aligned}$$



Radius versus diameter

Problems often give the diameter (pizzas, pipes, wheels); formulas mostly use the radius. Halve first. Also, πr^2 means $\pi(r^2)$, not $(\pi r)^2$: square the radius, then multiply by π. And keep π symbolic until the last step for an exact answer.

9.7.5. Circle theorems

Two facts about angles in circles have short, satisfying proofs and many uses.

Tangent–radius theorem. A tangent is perpendicular to the radius at the point of tangency. (Intuition: the radius is the shortest path from the centre to the tangent line, since every other point of the line is outside the circle; and the shortest path from a point to a line is the perpendicular one.)

Inscribed angle theorem. An inscribed angle is half the central angle standing on the same arc.

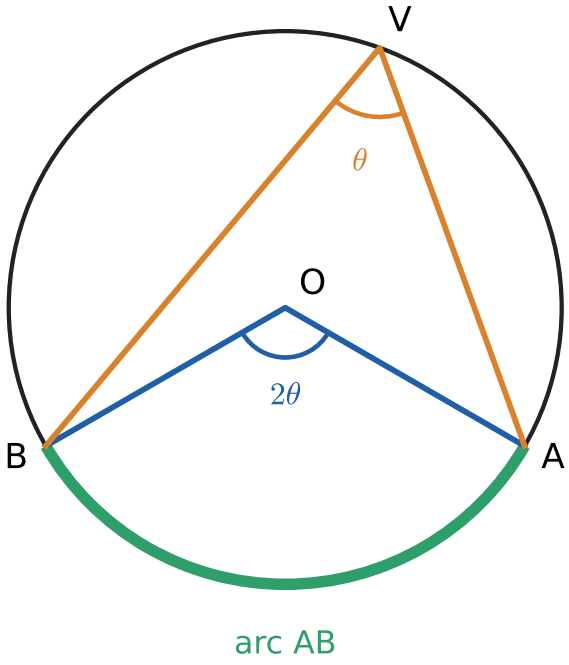


Figure 73. The inscribed angle at V is half the central angle at O on the same arc AB.

Proof (the case where the centre lies inside the inscribed angle). Draw the diameter from V through O. This splits everything into two halves; look at one. In triangle VOA, $OV = OA$ (both radii), so it is isosceles and its base angles at V and A are equal; call each x. The exterior angle at O of this triangle, the angle between $\overline{OA}$ and the extension of the diameter, is the sum of the two remote interior angles: $x + x = 2x$. The same argument on the other side gives an inscribed part y and a central part $2y$. So the inscribed angle is $x + y$ and the central angle is $2x + 2y = 2(x + y)$. ■ (The other cases, where the centre is outside the angle or on one side, use subtraction instead of addition.)

Consequences:

- **The stage puzzle.** Every inscribed angle standing on the same arc is equal, since each is half the same central angle. It doesn't matter where on the (major) arc you stand.
- **Thales' theorem.** An angle inscribed in a semicircle is a right angle, because the central angle is a straight line, $180°$, and half of that is $90°$. Practical use: to find the centre of a circular table, put the corner of a sheet of paper on the edge; the two points where the paper's edges cross the rim are the ends of a diameter. Do it twice; the diameters cross at the centre. *Try it:* See Thales' right angle .
- **Cyclic quadrilaterals** (all four vertices on a circle): opposite angles add to $180°$, since the two arcs they stand on make up the whole circle. *Try it:* Try a cyclic quadrilateral .

9.7.6. Where you'd use this

- **Driving:** a speedometer really counts wheel turns and multiplies by the circumference. Fit tyres 65 cm in diameter where the car expects 63 cm and each turn covers $65/63 \approx 1.032$ times as far: when the dial says 100 km/h you are doing about 103 km/h.
- **Navigation:** the Earth's radius is about 6371 km, so one degree of latitude is an arc of $\frac{1}{360} \cdot 2\pi \cdot 6371 \approx 111.2$ km, and one minute of arc is $111.2/60 \approx 1.853$ km. That is where the **nautical mile** (1852 m) comes from.
- **Plumbing and medicine:** the flow a pipe or blood vessel can carry depends on its cross-section πr^2, so doubling a pipe's diameter gives four times the area. (For slow, smooth flow it's even more dramatic: flow goes like r^4, which is why a slightly narrowed artery matters so much.)
- **Computing hardware:** a 7200 rpm hard disk spins 120 times a second. A track 4.5 cm from the centre has circumference $2\pi(0.045) \approx 0.283$ m, so it passes the head at about $0.283 \times 120 \approx 33.9$ m/s, more than twice as fast as a track near the hub. That's why the outer tracks read data fastest.



Computing pi

Archimedes (c. 250 BCE) trapped π between the perimeters of inscribed and circumscribed 96-sided polygons and proved $3\frac{10}{71} < \pi < 3\frac{1}{7}$, i.e. $3.1408 < \pi < 3.1429$. In 2024 a record calculation reached over 200 trillion digits. You need only about 40 digits to compute the circumference of the observable universe to within the width of a hydrogen atom; NASA's interplanetary navigation uses 15 or 16. Calculating π to huge precision is mostly a stress test for hardware and for fast multiplication algorithms.

Computer-science connection: circles on a grid

"Is point (x, y) inside a circle of radius r centred at (a, b)?" is the test $(x-a)^2 + (y-b)^2 \le r^2$, the Pythagorean theorem once more (again with no square root). It powers click-detection on round buttons, area-of-effect damage in games, and geofencing ("notify me within 200 m of the store", using a spherical version of distance for latitude and longitude). The same test with random points gives a Monte Carlo estimate of π: the fraction of random points in the unit square landing inside the quarter-circle approaches $\pi/4$. Chapter 10 treats circle equations and point-in-circle tests in coordinates.



Inscribed angle explorer

Fix points A and B on a circle and drag a third point V around the rim. The display shows $\angle AVB$ and the central angle $\angle AOB$ live. Watch $\angle AVB$ stay constant while V stays on one arc and jump to $180^\circ - \angle AVB$ when it crosses to the other. Drag A and B to opposite ends of a diameter to see Thales' right angle.

Try it: Open the inscribed angle explorer

US curriculum

Circumference and area of circles are **grade 7** (7.G.B.4). Arc length, sector area, and the inscribed-angle and tangent theorems belong to high-school **Geometry** (G-C.A–B).

9.7.7. Exercises

1. A bicycle wheel has diameter 70 cm. How far does the bike travel in 100 revolutions? Give an exact answer and a decimal in metres.
2. A sector of a circle of radius 9 has central angle 120°. Find its arc length and area, exactly.
3. An inscribed angle measures 35°. What is the central angle on the same arc? What is the measure of the arc?
4. AB is a diameter of a circle and C is another point on the circle with $AC = 6$, $BC = 8$. Find the radius.
5. A circle's area is 49π. Find its circumference.

▼ **Answers**

1. One revolution is $C = 70\pi$ cm; 100 revolutions is $7000\pi \text{ cm} = 70\pi \text{ m} \approx 219.9$ m.
2. Arc $= \frac{120}{360} \cdot 2\pi \cdot 9 = \frac{1}{3} \cdot 18\pi = 6\pi$. Area $= \frac{1}{3} \cdot 81\pi = 27\pi$.
3. Central angle 70°; an arc is measured by its central angle, so the arc is 70° too.
4. By Thales, $\angle ACB = 90^\circ$, so $AB = \sqrt{36 + 64} = 10$ and the radius is 5.

5. $\pi r^2 = 49\pi \Rightarrow r = 7$, so $C = 14\pi \approx 43.98$.

9.8. Surface area and volume

9.8.1. The problem

A soup can has radius 3.5 cm and height 10 cm. How much soup does it hold, and how much metal does it take to make? Why does a cone hold exactly a third of the cylinder that fits around it, not a half? And why is a sphere's surface area exactly four times the area of its shadow?



Solids

- A **prism** has two congruent parallel polygon **bases** joined by rectangular (or parallelogram) faces. A box is a rectangular prism; a cube has six square faces.
- A **cylinder** is the circular version of a prism.
- A **pyramid** has a polygon base and triangular faces meeting at a point, the **apex**. A **cone** is the circular version.
- A **sphere** is the set of points in space at distance r from a centre.
- **Volume** is the space enclosed, in cubic units. **Surface area** is the total area of all the faces or curved surfaces, in square units. **Lateral** surface area excludes the bases.
- The **height** h is always perpendicular to the base. A cone's **slant height** ℓ is measured along the surface from apex to rim: $\ell = \sqrt{r^2 + h^2}$.

9.8.2. The formulas and where they come from

Solid	Volume	Surface area
Prism (base area B, perimeter P)	$V = Bh$	$2B + Ph$
Cylinder	$V = \pi r^2 h$	$2\pi r^2 + 2\pi r h$
Pyramid	$V = \frac{1}{3} Bh$	base plus triangular faces
Cone	$V = \frac{1}{3}\pi r^2 h$	$\pi r^2 + \pi r \ell$
Sphere	$V = \frac{4}{3}\pi r^3$	$4\pi r^2$

Prisms and cylinders: stacking. A prism is a stack of identical thin slices, each with the base's area. Stack them to height h and you get $V = Bh$. A cylinder is a prism with a circular base, so $V = \pi r^2 h$. This "stack of slices" idea is **Cavalieri's principle**: two solids with equal cross-sectional areas at every height have equal volumes. (A stack of coins pushed sideways into a slanted pile keeps its volume.)

Cylinder surface area: unroll it. The curved side unrolls into a rectangle whose width is the circumference $2\pi r$ and whose height is h, area $2\pi rh$. Add two circular ends: $2\pi r^2 + 2\pi rh$.

Pyramids and cones: why one third? Here is a version you can hold in your hands. A cube of side s can be cut into **three identical square pyramids**, each with one face of the cube as its base and one shared corner of the cube as its apex: pick a corner, and the three faces of the cube that *don't* touch it each serve as the base of one pyramid. The three pyramids are congruent and fill the cube, so each has volume $\frac{1}{3}s^3 = \frac{1}{3} \times (\text{base}) \times (\text{height})$. Cavalieri's principle and stretching extend this to every pyramid, and a cone is a pyramid with so many sides that its base is a circle, giving $\frac{1}{3}\pi r^2 h$.

Another way to feel the third: a pyramid's cross-section shrinks as you go up, and its area is proportional to the **square** of the remaining distance to the apex. Adding up slices whose areas grow like t^2 gives the "one third" for the same reason that $\int_0^1 t^2\,dt = \frac{1}{3}$, as Volume 2, Chapter 19 will show.

Cone surface area. Cut the curved surface along a slant line and flatten it into a sector of a circle of radius ℓ whose arc length is the base circumference $2\pi r$. A sector's area is proportional to its arc: it is the fraction $\frac{2\pi r}{2\pi \ell}$ of the full disc $\pi\ell^2$, giving $\pi r\ell$.

The sphere. Archimedes considered this result his finest; he asked for it to be engraved on his tomb. Put a sphere of radius r snugly in a cylinder of radius r and height $2r$. Then:

- The sphere's volume is $\frac{2}{3}$ of the cylinder's: $\frac{2}{3} \cdot \pi r^2 \cdot 2r = \frac{4}{3}\pi r^3$.
- The sphere's surface area equals the cylinder's curved area: $2\pi r \cdot 2r = 4\pi r^2$. (Remarkably, this holds slice by slice: the area of any horizontal band of the sphere equals the area of the corresponding band of the cylinder. That is why the Lambert cylindrical equal-area map projection works.)
- The cone with the same base and height has volume $\frac{1}{3}\pi r^2 \cdot 2r = \frac{2}{3}\pi r^3$, and **cone** + **sphere** = **cylinder**: $\frac{2}{3}\pi r^3 + \frac{4}{3}\pi r^3 = 2\pi r^3$.

And the "four times its shadow" question: the sphere's shadow is a disc of area πr^2, and its surface is $4\pi r^2$.

9.8.3. Worked examples

The soup can ($r = 3.5$, $h = 10$, in cm). *Try it:* Cylinder volume and area .

$$V = \pi r^2 h = \pi(3.5)^2(10) = \pi(12.25)(10) = 122.5\pi \approx 384.8 \text{ cm}^3$$
$$SA = 2\pi r^2 + 2\pi rh = 2\pi(12.25) + 2\pi(3.5)(10) = 24.5\pi + 70\pi = 94.5\pi \approx 296.9 \text{ cm}^2$$

Since $1 \text{ cm}^3 = 1$ ml, the can holds about 385 ml.

An ice-cream cone with radius 3 cm and height 12 cm: *Try it:* Cone volume .

$$V = \tfrac{1}{3}\pi r^2 h = \tfrac{1}{3}\pi(9)(12) = 36\pi \approx 113.1 \text{ cm}^3.$$

A ball of radius 6 cm: $V = \frac{4}{3}\pi(216) = 288\pi \approx 904.8 \text{ cm}^3$ and $SA = 4\pi(36) = 144\pi \approx 452.4 \text{ cm}^2$. *Try it:* Sphere volume and area .



Slant height versus height

Volume formulas use the **perpendicular** height h; the cone's curved-surface formula uses the **slant** height ℓ. If you're given one and need the other, use $\ell^2 = r^2 + h^2$. And remember units: volume in cubic units, area in square units. Doubling every dimension of a box multiplies its volume by 8, not 2.

9.8.4. Where you'd use this

- **Home:** a round paddling pool 3 m across, filled 40 cm deep, holds $\pi(1.5)^2(0.4) \approx 2.83$ m³ ≈ 2830 litres. A garden hose delivering 12 litres a minute takes about $2830/12 \approx 236$ minutes, nearly four hours, to fill it.
- **Packaging:** manufacturers minimise surface area (material cost) for a given volume. For a closed cylinder that happens when the height equals the diameter, a classic optimisation problem you'll solve in Volume 2, Chapter 18 .
- **Biology and medicine:** small animals lose heat faster than large ones because surface area (heat loss) scales like k^2 while volume (heat production) scales like k^3; crushed ice melts faster than cubes for the same reason. Doctors track a roughly spherical tumour by its diameter, and a diameter that goes from 1 cm to 2 cm means $2^3 = 8$ times the volume.
- **Computing:** a 3D scan stored as a $512 \times 512 \times 512$ grid of one-byte **voxels** (volume pixels) takes 512^3 bytes $= 128$ MiB; doubling the resolution in each direction multiplies that by $2^3 = 8$, to 1 GiB. A 3D printer's slicer estimates the plastic needed from volumes: a solid 4 cm cube of PLA (density 1.24 g/cm³) is $64 \times 1.24 \approx 79$ g. Data centres care about the surface-to-volume ratio for cooling.

Computer-science connection: volumes in high dimensions

The volume of a "ball" generalises to any dimension, and something strange happens: as the dimension grows, almost all the volume of a ball concentrates in a thin shell near its surface, and the ball occupies a vanishing fraction of the cube around it (in 10 dimensions, about 0.25%). This is one face of the **curse of dimensionality**: random points in high-dimensional space are almost all far apart and nearly equidistant, which is why nearest-neighbour search on embeddings needs clever approximate indexes (HNSW, IVF) rather than geometric intuition from 3D.



Cube into three pyramids

A 3D model of a cube that splits along its interior diagonals into three congruent square pyramids, which can be dragged apart and rotated, then reassembled. A second panel fills a cone and a cylinder of the same base and height with "water" to show three cones fill one cylinder.

Try it: Cylinder: three cones' volume

US curriculum

Volume of prisms is **grade 5–6**; cylinders, cones and spheres are **grade 8** (8.G.C.9); surface area appears in **grade 6–7**. Informal arguments for the formulas, including Cavalieri's principle, are in high-school **Geometry** (G-GMD.A).

9.8.5. Exercises

1. Find the volume and total surface area of a box measuring 4 by 5 by 6.
2. A cylindrical tank has diameter 2 m and height 3 m. How many litres does it hold? ($1\ \text{m}^3 = 1000$ litres.)
3. A cone has radius 5 and height 12. Find its slant height, volume and total surface area, exactly.
4. A sphere has volume 36π. Find its radius and surface area.

▼ **Answers**

1. $V = 120$; $SA = 2(4\cdot5 + 4\cdot6 + 5\cdot6) = 2(20+24+30) = 148$.
2. $r = 1$: $V = \pi \cdot 1^2 \cdot 3 = 3\pi \approx 9.425\ \text{m}^3 \approx 9425$ litres.
3. $\ell = \sqrt{25 + 144} = 13$. $V = \tfrac{1}{3}\pi(25)(12) = 100\pi$.
   $SA = \pi(25) + \pi(5)(13) = 25\pi + 65\pi = 90\pi$.
4. $\tfrac{4}{3}\pi r^3 = 36\pi \Rightarrow r^3 = 27 \Rightarrow r = 3$; $SA = 4\pi\cdot 9 = 36\pi$.

9.9. Geometric proof as a way of reasoning

9.9.1. The problem

You've now seen several proofs in passing. What exactly makes an argument a proof? And why does the US Geometry course spend weeks having students write proofs in a rigid two-column format? The answer matters beyond geometry: a proof is a chain where each step is justified by something already accepted, which is the same discipline as reasoning about a program's correctness, writing a loop invariant, or arguing that a distributed protocol can't deadlock.

9.9.2. Intuition

A proof is a conversation with a sceptic who accepts only the axioms, definitions and previously proved theorems. Each step must be one the sceptic can't refuse. "It looks true in the diagram" is not a step. "These angles are vertical angles, so they're equal" is.

Proof vocabulary

- **Given**: the hypotheses you may assume.
- **Prove**: the conclusion you must reach.
- A **conditional** statement has the form "if P, then Q" ($P \Rightarrow Q$). P is the **hypothesis**, Q the **conclusion**.
- The **converse** of "if P then Q" is "if Q then P". It may or may not be true: "if a quadrilateral is a square, it has four equal sides" is true, but its converse is false (a rhombus).
- A **counterexample** is a single case showing a general statement is false.
- **Deductive** reasoning goes from general rules to specific conclusions (proof). **Inductive** reasoning (in the everyday sense, not mathematical induction) generalises from examples (conjecture).

Volume 3, Chapter 21 and Volume 3, Chapter 22 treat all of this formally.

9.9.3. The method: the two-column proof

Example 58. Procedure — writing a two-column proof

Input: a "Given" and a "Prove", and usually a diagram.

Steps:

1. Mark the diagram with everything given (tick marks for equal sides, arcs for equal angles, arrows for parallel lines).
2. Work backwards from the "Prove": what would let you conclude it? (Often: two triangles that would be congruent, giving the result by CPCTC.) Then what would establish **that**?
3. Write two columns: **Statements** on the left, **Reasons** on the right. The first statements are the givens (reason: "Given").
4. Each later statement must follow from earlier ones, with its reason being a definition, postulate, or previously proved theorem.
5. The last statement is the "Prove".

Output: a proof in which every step is justified.

Example 1: vertical angles are congruent.

Given: lines ℓ and m intersect, forming $\angle 1$, $\angle 2$, $\angle 3$ with $\angle 1$ and $\angle 3$ vertical.
Prove: $\angle 1 \cong \angle 3$.

#	Statement	Reason
1	ℓ and m intersect; $\angle 1, \angle 2$ and $\angle 2, \angle 3$ are linear pairs	Given (and definition of linear pair)
2	$m\angle 1 + m\angle 2 = 180^\circ$	Linear pair postulate (a linear pair is supplementary)
3	$m\angle 2 + m\angle 3 = 180^\circ$	Linear pair postulate
4	$m\angle 1 + m\angle 2 = m\angle 2 + m\angle 3$	Substitution (both equal 180°)
5	$m\angle 1 = m\angle 3$	Subtraction property of equality
6	$\angle 1 \cong \angle 3$	Definition of congruent angles

Example 2: the base angles of an isosceles triangle are equal.

Given: $\triangle ABC$ with $\overline{AB} \cong \overline{AC}$.
Prove: $\angle B \cong \angle C$.

Plan (working backwards): if we can split the triangle into two congruent halves, one containing $\angle B$ and the other $\angle C$, CPCTC finishes the job. Draw the bisector of $\angle A$, meeting $\overline{BC}$ at D.

#	Statement	Reason
1	$\overline{AB} \cong \overline{AC}$	Given
2	Draw $\overrightarrow{AD}$ bisecting $\angle BAC$, with D on $\overline{BC}$	Every angle has a unique bisector
3	$\angle BAD \cong \angle CAD$	Definition of angle bisector
4	$\overline{AD} \cong \overline{AD}$	Reflexive property (a segment is congruent to itself)
5	$\triangle ABD \cong \triangle ACD$	SAS (steps 1, 3, 4)
6	$\angle B \cong \angle C$	CPCTC

As a bonus, CPCTC also gives $BD = CD$ and $\angle ADB = \angle ADC$; since those two angles form a linear pair, each is 90°. So the bisector of the vertex angle is also the perpendicular bisector of the base.

9.9.4. Alternative styles

Mathematicians rarely write two-column proofs. The usual form is a **paragraph proof**, which is the same chain of justified steps in prose, as in the angle-sum and inscribed-angle proofs earlier in this chapter. There are also **flow-chart proofs** (boxes connected by arrows, popular in some US textbooks) and **coordinate proofs**, which place the figure on axes and compute (Chapter 10). Choosing: two-column format is a training tool that makes every justification explicit, and it's what US tests expect; paragraph proofs are what you'll read in every other mathematics book; coordinate proofs are best when the claim is about lengths, midpoints or slopes and a clever synthetic argument isn't obvious.



Proving the converse by accident

The most common logical error is using the statement you're trying to prove (circular reasoning), and the second most common is proving the converse instead. "If two lines are parallel then alternate angles are equal" and "if alternate angles are equal then the lines are parallel" are different theorems, each needing its own justification. Also, a diagram is never a reason: if a step says "because it looks perpendicular", the proof is broken.

Computer-science connection: proofs, invariants and verified geometry

A two-column proof is a program trace in which every line is checked against a rule. That analogy is literal in **proof assistants** such as Lean and Coq, where each step must be justified by a named lemma and the machine checks it; Book I of Euclid's **Elements** has been formalised this way, and the process found gaps in Euclid's original reasoning (he relied on diagrams for facts about points lying "between" others, which David Hilbert patched with extra axioms in 1899). When you argue that a loop maintains an invariant, you are writing a paragraph proof (Volume 3, Chapter 22). And there's a practical lesson for geometry code: floating-point arithmetic doesn't respect the axioms exactly, so the orientation test "do these three points turn left?" can give inconsistent answers for nearly collinear points, a famous source of crashes in convex-hull and mesh code. Robust libraries use exact or adaptive-precision arithmetic for these predicates.



Euclid's best seller

Euclid's **Elements**, thirteen books and 465 propositions, was the standard geometry textbook for more than two thousand years and is said to have gone through more editions than any book except the Bible. Abraham Lincoln studied it as a young lawyer to learn what it meant to "demonstrate" something, working through the first six books. The two-column format is much newer: it was popularised in US high schools around 1900–1910.

US curriculum

Formal proof (two-column, paragraph and flow-chart) is the signature content of high-school **Geometry**, usually the only US course where students write proofs. Proofs about lines, angles, triangles and parallelograms are Common Core G-CO.C.9–11.

9.9.5. Exercises

1. State the converse of "If a triangle is equilateral, then it is isosceles." Is the converse true?
2. Give a counterexample to: "Every quadrilateral with four equal sides is a square."
3. Write a two-column proof: Given $\angle 1$ and $\angle 2$ are both complementary to $\angle 3$. Prove $\angle 1 \cong \angle 2$.
4. Write a paragraph proof that the diagonals of a parallelogram bisect each other. (Hint: alternate interior angles and ASA.)

▼ **Answers**

1. "If a triangle is isosceles, then it is equilateral." False: a triangle with sides 5, 5, 8 is isosceles but not equilateral.
2. A rhombus with angles 60° and 120°.
3. (1) $\angle 1$ and $\angle 3$ are complementary; $\angle 2$ and $\angle 3$ are complementary (Given). (2) $m\angle 1 + m\angle 3 = 90^\circ$ and $m\angle 2 + m\angle 3 = 90^\circ$ (definition of complementary). (3) $m\angle 1 + m\angle 3 = m\angle 2 + m\angle 3$ (substitution). (4) $m\angle 1 = m\angle 2$ (subtraction property). (5) $\angle 1 \cong \angle 2$ (definition of congruent angles).
4. Let parallelogram $ABCD$ have diagonals meeting at M. Since $\overline{AB} \parallel \overline{DC}$, the transversal $\overline{AC}$ gives $\angle MAB = \angle MCD$ and the transversal $\overline{BD}$ gives $\angle MBA = \angle MDC$ (alternate interior angles). Opposite sides of a parallelogram are equal, so $AB = CD$. By ASA, $\triangle MAB \cong \triangle MCD$. By CPCTC, $MA = MC$ and $MB = MD$, so M is the midpoint of both diagonals.

9.10. Choosing a tool: a summary

You want …	Reach for …
an unknown angle at a crossing or with parallel lines	vertical / linear-pair / corresponding / alternate angle facts
an unknown angle in a triangle or polygon	angle sum $(n-2)\cdot 180°$, exterior angles sum to $360°$
an unknown length, two triangles with matching angles	similarity (AA), then a proportion
an unknown length in a right triangle	Pythagoras (or special triangles; later, trigonometry)
to show two triangles are identical	SSS, SAS, ASA, AAS, HL (never SSA or AAA)
the distance between two points	$\sqrt{\Delta x^2+\Delta y^2}$; compare squared distances in code
the area of a polygon given by coordinates	shoelace formula
lengths, areas or volumes after scaling by k	k, k^2, k^3
an angle subtended in a circle	inscribed angle = half the central angle; semicircle gives $90°$

9.11. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Geometry: triangles, circles, polygons, solids
- Conics, pixels and metrics

9.12. Mastery check

1. Two parallel lines are cut by a transversal. One of the angles formed is $(4x - 10)°$ and its same-side interior partner is $(2x + 40)°$. Find both angles.
2. A regular polygon has interior angles of $140°$. How many sides does it have, and what is the sum of its interior angles?
3. A 1.5 m fence post casts a 2 m shadow; at the same moment a flagpole casts a 14 m shadow. How tall is the flagpole? Name the similarity criterion you used.
4. A rectangular screen is 24 inches wide and 10 inches tall. How long is its diagonal? Is a triangle with sides 12, 35, 37 a right triangle?
5. Find the area of the polygon with vertices $(0, 0)$, $(4, 0)$, $(4, 4)$, $(2, 1)$, $(0, 4)$ (in that order), using the shoelace formula. Is it convex?
6. A circle has radius 10. Find the length of an arc and the area of a sector with central angle $72°$, exactly. An inscribed angle stands on the same arc; what is it?
7. A cone and a cylinder have the same radius 3 and height 8. Find both volumes and their ratio. A sphere of radius 3 is scaled by 2; by what factor do its surface area and its volume change?
8. Explain in one or two sentences why SSA is not a valid congruence criterion, and write a two-column proof that if M is the midpoint of $\overline{AB}$ and of $\overline{CD}$ then $\triangle AMC \cong \triangle BMD$.

Decision rule: All correct without hesitation → move on to Chapter 10, where every fact here becomes algebra. Stuck on 1–2 → reread Section 9.2. Stuck on 3–4 → reread Section 9.4 and Section 9.5. Stuck on 5 → Section 9.6.4. Stuck on 6–7 → Section 9.7 and Section 9.8. Stuck on 8 → Section 9.9. Stuck on more than three → work the section exercises first.

▼ **Mastery check answers**

1. Same-side interior angles are supplementary: $(4x - 10) + (2x + 40) = 180 \Rightarrow 6x + 30 = 180 \Rightarrow x = 25$. The angles are $4(25) - 10 = 90°$ and $2(25) + 40 = 90°$: the transversal is perpendicular.
2. Exterior angle $180 - 140 = 40°$, so $n = 360/40 = 9$ (a nonagon). Sum $(9 - 2) \times 180 = 1260°$. Check: $9 \times 140 = 1260$.
3. AA similarity (right angle and the sun's elevation angle): $h/1.5 = 14/2 = 7$, so $h = 10.5$ m.
4. $\sqrt{24^2 + 10^2} = \sqrt{576 + 100} = \sqrt{676} = 26$ inches. $12^2 + 35^2 = 144 + 1225 = 1369 = 37^2$, so yes, by the converse of the Pythagorean theorem.
5. Terms: $0 \cdot 0 - 4 \cdot 0 = 0$; $4 \cdot 4 - 4 \cdot 0 = 16$; $4 \cdot 1 - 2 \cdot 4 = -4$; $2 \cdot 4 - 0 \cdot 1 = 8$; $0 \cdot 0 - 0 \cdot 4 = 0$. Sum 20, area 10. Check: the 4×4 square (16) minus the notch triangle $(0, 4), (2, 1), (4, 4)$ with base 4 and height 3 (area 6) gives 10. It is **concave**: the interior angle at $(2, 1)$ exceeds $180°$.

6. Arc $= \frac{72}{360} \cdot 2\pi \cdot 10 = \frac{1}{5} \cdot 20\pi = 4\pi$. Sector $= \frac{1}{5} \cdot 100\pi = 20\pi$. Inscribed angle $= 72/2 = 36°$.

7. Cylinder $\pi \cdot 9 \cdot 8 = 72\pi$; cone $\frac{1}{3} \cdot 72\pi = 24\pi$; ratio $1:3$. Scaling by 2: surface area ×4, volume ×8.

8. SSA can produce two different triangles: the side opposite the given angle can swing to meet the third side at two points. Proof: (1) M is the midpoint of $\overline{AB}$ and of $\overline{CD}$ (Given). (2) $AM = MB$ and $CM = MD$ (definition of midpoint). (3) $\angle AMC \cong \angle BMD$ (vertical angles). (4) $\triangle AMC \cong \triangle BMD$ (SAS, using 2 and 3; the angle is between the two sides).

Chapter 10. Coordinate Geometry

Draw a grid, give every point an address, and geometry becomes algebra. This chapter rebuilds the coordinate plane from scratch: distance and midpoint (Pythagoras in disguise), the three standard equations of a line, the slope test for perpendicular lines, circles by completing the square, a tour of the conic sections, and the four basic transformations written as coordinate rules. It ends where computer graphics begins: pixel grids with y pointing down, Bresenham's line algorithm, bounding boxes, collision tests, and the difference between Euclidean and Manhattan distance.

Why it matters: every map app, sat-nav, CAD drawing, game, plotting library and GPS receiver works by storing positions as numbers and doing algebra on them. Once you can turn "is the phone within range of the mast?" or "where do these two roads cross?" into an equation, a computer can answer it a billion times a second.

Where this chapter sits

You need: signed numbers and squares and square roots (Chapter 1, Chapter 3); rearranging equations (Chapter 4, Chapter 5), especially slope and $y = mx + b$ from Chapter 5; completing the square (Chapter 8); and the Pythagorean theorem, congruence and similarity (Chapter 9).

This unlocks: graphs of functions (Chapter 11); the unit circle and rotations (Chapter 13); vectors, which are coordinates with a direction attached (Volume 2, Chapter 26); and transformations as matrices (Volume 2, Chapter 27). Calculus (Volume 2, Chapter 17) is, at heart, the study of slope on curved graphs drawn on exactly this plane.

10.1. Descartes' idea: geometry becomes algebra

10.1.1. The problem

Here is a classical geometry question: *the three medians of a triangle (each joining a vertex to the midpoint of the opposite side) always meet at a single point. Why?* With ruler-and-compass reasoning this takes a clever argument. With coordinates, it is a few lines of arithmetic: write the vertices as (x_1, y_1), (x_2, y_2), (x_3, y_3), compute the point two-thirds of the way along each median, and notice that all three computations give the same answer, $\left(\frac{x_1+x_2+x_3}{3}, \frac{y_1+y_2+y_3}{3}\right)$.

The reverse direction is just as powerful. The equation $x^2 + y^2 = 25$ is a dry piece of algebra until you see that its solutions trace out a circle of radius 5. A picture makes the algebra obvious; algebra makes the picture precise.

10.1.2. Intuition

The idea, usually credited to René Descartes (and independently to Pierre de Fermat) in the 1630s, is to give every point an *address*: two numbers that say how far to go across and how far to go up from a fixed starting point. Once points are numbers:

- a **shape** becomes an **equation** (the set of addresses that satisfy it);
- a **geometric question** ("do these lines cross?") becomes an **algebra question** ("does this system have a solution?"), which you already know how to answer from Chapter 6;
- a **geometric operation** ("rotate this triangle") becomes a **formula** applied to each address.

That third bullet is the foundation of every graphics engine, map application and CAD program ever written. A computer cannot see a triangle; it can only store and transform coordinates.



Analytic geometry

Analytic geometry (also **coordinate geometry** or **Cartesian geometry**) is the study of geometry using a coordinate system, so that geometric facts are proved and computed with algebra. It contrasts with **synthetic geometry** — Euclid's style in Chapter 9 — which reasons directly from axioms and diagrams without numbers.



The fly on the ceiling

A much-repeated story says Descartes, lying in bed, watched a fly crawl across the ceiling and realised he could describe its position at every moment by its distances from two walls. The story is almost certainly a legend, but the idea is right. Descartes published it in 1637 as *La Géométrie*, an appendix to his *Discourse on the Method*. The word **Cartesian** comes from his Latinised name, *Renatus Cartesius*. Fermat had the same idea at about the same time but did not publish it, a recurring theme in Fermat's career.

10.2. The Cartesian plane

10.2.1. Vocabulary and notation

Take two perpendicular number lines that cross at their zeros. The horizontal one is the **x-axis**, with positive numbers to the right; the vertical one is the **y-axis**, with positive numbers up. Every point P in the plane now has a unique address (x, y): drop a perpendicular from P to each axis and read off the numbers.



Cartesian plane, axes, origin

The **Cartesian plane** (or **coordinate plane**, **xy-plane**) is the plane together with two perpendicular number lines, the **axes**. They meet at the **origin**, $O = (0, 0)$. The horizontal axis is the **x-axis**; the vertical axis is the **y-axis**.



Coordinates, ordered pair

A point's **coordinates** are the **ordered pair** (x, y). The first number, the **x-coordinate** (old name: **abscissa**), is the signed horizontal distance from the y-axis. The second, the **y-coordinate** (old name: **ordinate**), is the signed vertical distance from the x-axis. "Ordered" matters: $(2, 5)$ and $(5, 2)$ are different points.

Quadrant

The axes cut the plane into four **quadrants**, numbered with Roman numerals **anticlockwise** starting from the top right: **I** (x positive, y positive), **II** (x negative, y positive), **III** (both negative), **IV** (x positive, y negative). Points that lie **on** an axis belong to no quadrant.

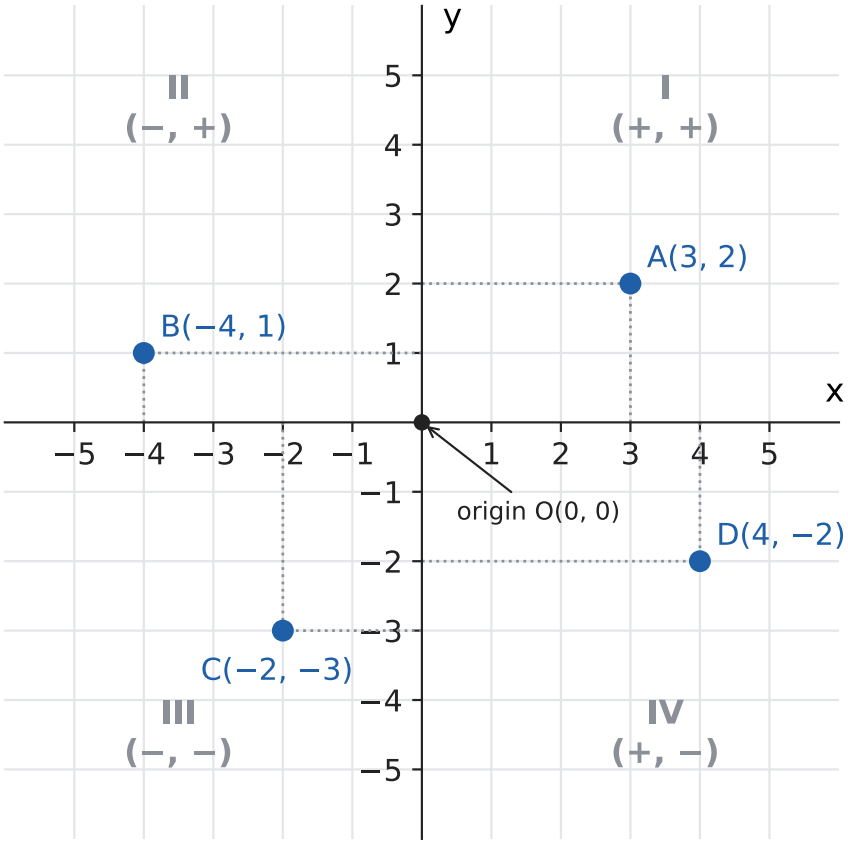


Figure 74. The coordinate plane. Each point's address is read by dropping perpendiculars to the axes.

The anticlockwise numbering is not arbitrary: it is the direction in which a positive angle turns, which you will see again with the unit circle in Chapter 13.

(x, y), not (y, x) — and not (row, column)

The order is always "across, then up": x first. A helpful mnemonic is that you walk along the corridor before you go up the stairs. Programmers have an extra trap: two-dimensional arrays are indexed `grid[row][col]`, which is `grid[y][x]` — the opposite order. Mixing the two is one of the most common bugs in image-processing and game code.

US curriculum

Plotting points in the first quadrant appears in **grade 5**; all four quadrants in **grade 6**. In **grade 8** students use the Pythagorean theorem to find distances between points, and reflections, rotations, translations and dilations on the coordinate plane are a major Common Core grade-8 topic. **Algebra I** develops lines; high-school **Geometry** uses coordinates to prove theorems ("coordinate proofs"); **Algebra II** and **Precalculus** cover circles and conic sections.

10.2.2. Exercises

1. Name the quadrant (or axis) containing each point: $(-3, 5)$, $(0, -4)$, $(2, -7)$, $(-1, -1)$.
2. A point has a negative x-coordinate and its y-coordinate is twice its x-coordinate. Which quadrant is it in?
3. The point (a, b) is in quadrant II. Which quadrant contains (b, a)? Which contains $(-a, b)$?

▼ **Answers**

1. $(-3, 5)$: quadrant II. $(0, -4)$: on the y-axis (no quadrant). $(2, -7)$: quadrant IV. $(-1, -1)$: quadrant III.
2. If $x < 0$ then $y = 2x < 0$ as well, so both coordinates are negative: quadrant III.
3. In quadrant II, $a < 0$ and $b > 0$. Then (b, a) has positive x and negative y: quadrant IV. And $(-a, b)$ has $-a > 0$ and $b > 0$: quadrant I (it is the mirror image in the y-axis).

10.3. Distance and midpoint

10.3.1. The problem

A delivery drone sits at $(-2, 1)$ on a city map marked in kilometres; the customer is at $(4, 9)$. How far is the straight-line flight? And where should a relay station go if it must be exactly halfway?

10.3.2. Intuition

Any two points that are not level with each other are the ends of the hypotenuse of a right triangle whose other two sides run parallel to the axes. The horizontal leg is the change in x; the vertical leg is the change in y. Pythagoras (from Chapter 9) does the rest.

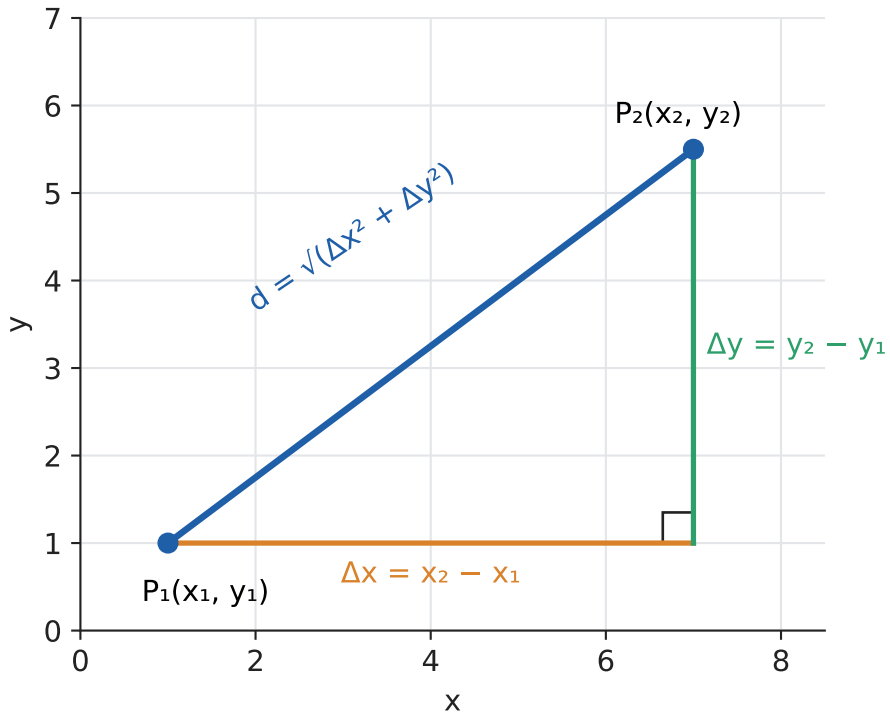


Figure 75. The distance between two points is the hypotenuse of a right triangle with legs Δx and Δy.



Delta notation

The Greek capital letter **delta**, Δ, is read "change in". So $\Delta x = x_2 - x_1$ is the change in x going from the first point to the second. The subscripts x_1, x_2 are just labels ("x of point 1", "x of point 2"), not powers or multiplications.

10.3.3. Why it works

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. The corner point (x_2, y_1) makes a right angle, because one leg is horizontal and the other vertical. The legs have lengths $|x_2 - x_1|$ and $|y_2 - y_1|$. By Pythagoras:

$$
\begin{aligned}
d^2 &= |x_2 - x_1|^2 + |y_2 - y_1|^2 && \text{(Pythagoras)} \\
&= (x_2 - x_1)^2 + (y_2 - y_1)^2 && \text{(squaring removes the sign, so the bars can go)} \\
d &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} && \text{(take the positive root: a distance)}
\end{aligned}
$$

Two pleasant consequences fall out. Because each difference is squared, it does not matter which point you call "1": $(x_1 - x_2)^2 = (x_2 - x_1)^2$. And if the points happen to be level ($y_1 = y_2$), the formula still works: it collapses to $\sqrt{(x_2 - x_1)^2} = |x_2 - x_1|$, which is the ordinary number-line distance.



Distance formula

The **distance formula**: the distance between (x_1, y_1) and (x_2, y_2) is $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. It is the Pythagorean theorem written in coordinates. This straight-line distance is called the **Euclidean distance**.

The midpoint is even simpler. Halfway between two numbers on a line is their average; halfway between two points is the average in each coordinate separately, because moving halfway along the hypotenuse moves you halfway along each leg (the half-size triangle is similar to the full one, with scale factor $\frac{1}{2}$).



Midpoint formula

The **midpoint** of the segment from (x_1, y_1) to (x_2, y_2) is $M = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right)$, the coordinate-wise average. A **segment** (line segment) is the part of a line between two **endpoints**.

10.3.4. The method

Example 59. Procedure — distance and midpoint of two points

Input: two points $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$.

Steps:

1. Compute $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$ (watch the signs when subtracting negatives).
2. Square each: Δx^2, Δy^2.
3. Add them, then take the square root: $d = \sqrt{\Delta x^2 + \Delta y^2}$. Simplify the surd if exact form is wanted.
4. For the midpoint, average the x-coordinates and average the y-coordinates.

Output: the distance d and the midpoint M.

10.3.5. Worked example

Drone at $P_1 = (-2, 1)$, customer at $P_2 = (4, 9)$.

$$\Delta x = 4 - (-2) = 6 \qquad \Delta y = 9 - 1 = 8$$
$$d = \sqrt{6^2 + 8^2} = \sqrt{36 + 64} = \sqrt{100} = 10 \text{ km}$$
$$M = \left(\frac{-2 + 4}{2}, \frac{1 + 9}{2}\right) = \left(\frac{2}{2}, \frac{10}{2}\right) = (1, 5)$$

Sanity check: the 6–8–10 triangle is the familiar 3–4–5 triangle doubled. And the distance from $(-2, 1)$ to the midpoint $(1, 5)$ is $\sqrt{3^2 + 4^2} = 5$, exactly half. Good.

A second example with an unfriendly answer: from $(1, -3)$ to $(-4, 2)$.

$$\Delta x = -4 - 1 = -5 \qquad \Delta y = 2 - (-3) = 5$$
$$d = \sqrt{(-5)^2 + 5^2} = \sqrt{50} = \sqrt{25 \cdot 2} = 5\sqrt{2} \approx 7.07$$

10.3.6. Beyond the midpoint: any fraction of the way

What if you want the point one quarter of the way from P_1 to P_2? The same similar-triangles argument says: move a quarter of Δx and a quarter of Δy. In general, the point a fraction t of the way along is

$$P(t) = \big(x_1 + t\,(x_2 - x_1),\; y_1 + t\,(y_2 - y_1)\big), \qquad 0 \le t \le 1.$$

At $t = 0$ you are at P_1; at $t = 1$ at P_2; at $t = \frac{1}{2}$ at the midpoint. This is the **parametric form** of a segment: the parameter t acts like time, and $P(t)$ is where you are at that time. Equivalently, $P(t) = (1 - t)P_1 + tP_2$, a **weighted average** of the endpoints.

Computer-science connection: lerp

Graphics and game programmers call $P(t) = (1 - t)P_1 + tP_2$ **lerp** (linear interpolation). It animates objects from one place to another (increase t each frame), blends colours (treat (r, g, b) as coordinates), and is the building block of Bézier curves, which are lerps of lerps. Shader languages provide it as `mix(a, b, t)`. Note the form `(1 - t) * a + t * b` is numerically safer than `a + t * (b - a)`: it returns exactly `b` at $t = 1$ in floating point.

10.3.7. Choosing a method, and implementation

By hand, compute the differences first, then square; it keeps signs straight. In code, the standard library already knows the formula.

PYTHON

```python
import math

def distance(p, q):
    """Euclidean distance between points p=(x1, y1) and q=(x2, y2)."""
    dx = q[0] - p[0]
    dy = q[1] - p[1]
    return math.sqrt(dx*dx + dy*dy)      # or math.hypot(dx, dy)

def midpoint(p, q):
    return ((p[0] + q[0]) / 2, (p[1] + q[1]) / 2)

def slope(p, q):
    dx = q[0] - p[0]
    if dx == 0:
        return None                      # vertical line: slope undefined
    return (q[1] - p[1]) / dx

P, Q = (-2, 1), (4, 9)
print(distance(P, Q), midpoint(P, Q), slope(P, Q))
print(math.dist(P, Q))                   # Python 3.8+, any dimension
```

```
10.0 (1.0, 5.0) 1.3333333333333333
10.0
```

`math.hypot` avoids overflow when Δx is enormous (squaring a 10^{200} overflows a double; `hypot` rescales internally). `math.dist` works in any number of dimensions, which you will need in Section 10.10.

10.3.8. Failure modes



Three classic distance-formula slips

- **Subtracting a negative.** $4 - (-2) = 6$, not 2. Write the brackets in.
- **Square root of a sum is not a sum of square roots.** $\sqrt{6^2 + 8^2} = 10$, not $6 + 8 = 14$. (14 is a different, also useful, distance — see Section 10.13.)
- **Midpoint is an average, not a difference.** $\frac{x_1 + x_2}{2}$, with a plus sign. $\frac{x_2 - x_1}{2}$ is half the *length* of the horizontal leg, not a coordinate.

10.3.9. Where you'd use this

- **Navigation:** on a small enough patch of the Earth that it looks flat, map coordinates plus the distance formula give straight-line distances. On a hiking map with a 1 km grid, a walk from grid position $(3.2, 7.5)$ to $(6.8, 2.7)$ is $\sqrt{3.6^2 + 4.8^2} = \sqrt{12.96 + 23.04} = \sqrt{36} = 6$ km as the crow flies; the path will be longer, never shorter. (On a global scale you need spherical geometry; GPS uses 3D coordinates, see Section 10.10.)
- **Coordinate proofs:** to prove a quadrilateral is a parallelogram, show its diagonals have the same midpoint; to prove a triangle is isosceles, show two side lengths are equal.
- **Machine learning:** the k-nearest-neighbours classifier labels a new point by looking at the k training points with the smallest Euclidean distance to it.

Computer-science connection: skip the square root

When you only need to *compare* distances ("which enemy is nearest?", "is this point within range?"), compare **squared** distances instead. Since both sides are non-negative, $d_1 < d_2 \iff d_1^2 < d_2^2$, and the square root — a relatively expensive operation — disappears. Collision code, nearest-neighbour searches and clustering inner loops all do this.



Drag the triangle

Drag two points around the plane. The app draws the right triangle with legs Δx and Δy, shows $\Delta x^2 + \Delta y^2 = d^2$ updating live, and marks the point $P(t)$ as you move a slider for t from 0 to 1. Try dragging one point across the other and watch Δx change sign while d stays positive.

10.3.10. Exercises

1. Find the distance and midpoint of $(1, 2)$ and $(4, 6)$.
2. Find the distance and midpoint of $(-3, -1)$ and $(2, 11)$.

3. The midpoint of AB is $M(3, -1)$ and $A = (-2, 4)$. Find B.
4. Find the point one quarter of the way from $(0, 0)$ to $(8, 12)$.
5. Show that the triangle with vertices $A(1, 1)$, $B(4, 3)$, $C(2, 6)$ is a right isosceles triangle. Where is the right angle?

▼ **Answers**

1. $\Delta x = 3$, $\Delta y = 4$, so $d = \sqrt{9 + 16} = 5$. Midpoint $\left(\frac{1+4}{2}, \frac{2+6}{2}\right) = (2.5, 4)$.
2. $\Delta x = 2 - (-3) = 5$, $\Delta y = 11 - (-1) = 12$, so $d = \sqrt{25 + 144} = \sqrt{169} = 13$. Midpoint $\left(\frac{-1}{2}, \frac{10}{2}\right) = (-0.5, 5)$.
3. M is the average, so $B = 2M - A$ coordinate-wise: $x_B = 2(3) - (-2) = 8$, $y_B = 2(-1) - 4 = -6$. $B = (8, -6)$.
4. With $t = \frac{1}{4}$: $(0 + \frac{1}{4} \cdot 8,\ 0 + \frac{1}{4} \cdot 12) = (2, 3)$.
5. Squared side lengths: $AB^2 = 3^2 + 2^2 = 13$, $BC^2 = (-2)^2 + 3^2 = 13$, $AC^2 = 1^2 + 5^2 = 26$. Two sides are equal (isosceles) and $AB^2 + BC^2 = 26 = AC^2$, so by the converse of Pythagoras the angle opposite AC — at B — is a right angle.

10.4. Slope revisited

You met slope in Chapter 5 as "rise over run". In coordinates it is

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}, \qquad x_1 \neq x_2.$$

Three facts deserve restating now that we have the full plane.



Slope

The **slope** (or **gradient**) m of a non-vertical line is $\frac{\Delta y}{\Delta x}$ between any two of its points — the change in y per unit increase in x. Positive slope rises left-to-right, negative slope falls, zero slope is horizontal, and a vertical line has **undefined** slope ($\Delta x = 0$, so you would divide by zero).

Slope is the same between any two points on a line. Pick two different pairs of points on one line; the run–rise triangles they make are similar (equal angles, from parallel sides), so their ratios agree. This is why "the slope of a line" is meaningful at all, and it gives a quick **collinearity test**: three points lie on one line exactly when the slope from the first to the second equals the slope from the second to the third (or both are vertical).

Slope measures steepness as a tangent. If a line makes an angle θ with the positive x-axis (its **angle of inclination**), then $m = \tan\theta$. You will meet $\tan$ properly in Chapter 13; for now, note that a 45° line has slope 1 and that slope grows without bound as the line approaches vertical.

Slope is a rate. If x is time in hours and y is distance in km, the slope is speed in km per hour. That reading — slope as rate of change — is exactly what calculus generalises to curves (Volume 2, Chapter 17).

Worked example: are $A(1,1)$, $B(3,4)$, $C(7,10)$ collinear? *Try it:* Check collinearity .

$$m_{AB} = \frac{4-1}{3-1} = \frac{3}{2}, \qquad m_{BC} = \frac{10-4}{7-3} = \frac{6}{4} = \frac{3}{2}.$$

The slopes are equal and the segments share point B, so yes. (Equal slopes alone are not enough: two *parallel* segments also have equal slopes. Sharing a point is what forces them onto one line.)

Where you'd use slope. Building regulations state slopes as ratios: a wheelchair ramp may be no steeper than 1:12 (US and UK guidance), so a 60 cm step needs at least $0.6 \times 12 = 7.2$ m of ramp run. A road sign warning of a 10% hill means a slope of 0.1: 10 m up for every 100 m along, an angle of only about $5.7°$. In mechanics the slope of a distance–time graph is speed and the slope of a velocity–time graph is acceleration: a car going from 0 to 27 m/s in 9 s has an average acceleration of $27/9 = 3$ m/s². In computing, the slope of a "requests served vs. time" graph is throughput, and the slope of a log of memory use is the leak rate you're hunting.

10.5. Equations of lines

10.5.1. The problem

You know two points a road passes through, or one point and its gradient. You need a single equation that every point on the road satisfies — so a program can test whether a given point is on the road, find where it crosses another road, or draw it. Which equation, and in which form?

10.5.2. Intuition

A line is pinned down by two pieces of information: either **two points**, or **one point and a slope**. Each standard form of the equation is just one of those descriptions, written as an equation in x and y.

10.5.3. Why it works: three forms

Point-slope form. Suppose the line has slope m and passes through (x_1, y_1). A point (x, y) (other than (x_1, y_1) itself) is on the line exactly when the slope from (x_1, y_1) to it equals m:

$$\frac{y - y_1}{x - x_1} = m \qquad \text{(slope from the fixed point to the moving point)}$$
$$y - y_1 = m\,(x - x_1) \qquad \text{(multiply both sides by } x - x_1\text{)}$$

The second line also holds at (x_1, y_1) (both sides are zero), so it describes the whole line.

Slope-intercept form. Take the special point where the line crosses the y-axis, $(0, b)$. Point-slope gives $y - b = m(x - 0)$, i.e.

$$y = mx + b.$$

Standard form. Move everything to one side and clear fractions:

$$Ax + By = C, \qquad A, B \text{ not both zero.}$$

Conventionally A, B, C are integers with no common factor and $A \geq 0$. Standard form is the only one of the three that can describe a **vertical** line: $x = 5$ is $1 \cdot x + 0 \cdot y = 5$. Its slope is $-A/B$ (when $B \neq 0$); its intercepts are easy to read: set $y = 0$ to get the x-intercept C/A, set $x = 0$ to get the y-intercept C/B.



Forms of a linear equation

- **Slope-intercept form:** $y = mx + b$ — slope m, **y-intercept** b (where the line crosses the y-axis).
- **Point-slope form:** $y - y_1 = m(x - x_1)$ — slope m through the known point (x_1, y_1).
- **Standard form:** $Ax + By = C$ — handles vertical lines; good for systems of equations and integer coefficients.

An **intercept** is where a graph crosses an axis: the **x-intercept** has $y = 0$, the **y-intercept** has $x = 0$.

10.5.4. The method

Example 60. Procedure — equation of a line

Input: either two points $(x_1, y_1), (x_2, y_2)$, or a point (x_1, y_1) and a slope m.

Steps:

1. If given two points with $x_1 = x_2$, stop: the line is vertical, $x = x_1$.
2. If given two points, compute $m = \frac{y_2 - y_1}{x_2 - x_1}$.
3. Write point-slope form $y - y_1 = m(x - x_1)$ using either point.
4. To get slope-intercept form, expand and add y_1 to both sides.
5. To get standard form, clear fractions (multiply by the denominators), then gather x and y terms on the left and the constant on the right; make $A \geq 0$.
6. Check: substitute the **other** given point (or any known point) into your final equation.

Output: the line's equation in the form you need.

10.5.5. Worked examples

Two points. Line through $(1, 2)$ and $(3, 8)$.

$$
\begin{aligned}
m &= \frac{8-2}{3-1} = \frac{6}{2} = 3 && \text{(slope)} \\
y - 2 &= 3(x-1) && \text{(point-slope, using } (1,2)) \\
y - 2 &= 3x - 3 && \text{(expand)} \\
y &= 3x - 1 && \text{(add 2: slope-intercept)} \\
3x - y &= 1 && \text{(rearrange: standard)}
\end{aligned}
$$

Check with $(3, 8)$: $3(3) - 1 = 8$. Correct.

Point and fractional slope. Through $(-4, 1)$ with slope $-\frac{2}{3}$.

$$
\begin{aligned}
y - 1 &= -\tfrac{2}{3}(x+4) && \text{(point-slope; } x-(-4) = x+4) \\
3(y-1) &= -2(x+4) && \text{(multiply both sides by 3)} \\
3y - 3 &= -2x - 8 && \text{(expand)} \\
2x + 3y &= -5 && \text{(add } 2x+3 \text{ to both sides: standard)}
\end{aligned}
$$

Slope-intercept is $y = -\frac{2}{3}x - \frac{5}{3}$. Check: $2(-4) + 3(1) = -8 + 3 = -5$. Correct.

From standard form to a sketch. $4x - 3y = 12$. Intercepts: $y = 0 \Rightarrow x = 3$; $x = 0 \Rightarrow y = -4$. Plot $(3, 0)$ and $(0, -4)$ and join them. Slope: $-A/B = -4/(-3) = \frac{4}{3}$.

10.5.6. Choosing a form

Form	Use it when…	Weakness
$y = mx + b$	graphing, reading off slope and intercept, computing y from x	cannot express vertical lines
$y - y_1 = m(x - x_1)$	**building** the equation from a point and a slope (the fastest route)	not unique: each point gives a different-looking equation
$Ax + By = C$	solving systems (Chapter 6), integer arithmetic, vertical lines, quick intercepts	slope not visible at a glance

Rule of thumb: **build** in point-slope, **present** in slope-intercept for graphing or in standard form for systems and code.

10.5.7. Where you'd use this

- **Deciding between deals.** A gym charges \$8 per visit, or \$30 a month plus \$2 per visit. Monthly cost is $y = 8v$ or $y = 30 + 2v$ for v visits; the lines cross where $8v = 30 + 2v$, i.e. $v = 5$. Five visits a month costs \$40 either way; more than five and the membership wins. Phone plans, car hire and electricity tariffs (standing charge plus a rate per kWh) are all lines with an intercept and a slope.
- **Unit conversion.** Fahrenheit is a linear function of Celsius. Two known points, $(0, 32)$ (water freezes) and $(100, 212)$ (water boils), give slope $180/100 = 1.8$ and the line $F = 1.8C + 32$.

- **Calibrating a sensor.** A temperature sensor outputs 0.50 V at 20 °C and 1.30 V at 100 °C. Its slope is $(1.30 - 0.50)/(100 - 20) = 0.01$ V per °C, so a reading of 0.82 V means $T = 20 + (0.82 - 0.50)/0.01 = 52$ °C. Every two-point calibration in a lab or in firmware is "the line through two points". Hooke's law for a spring ($F = kx$) is the same idea with intercept zero.
- **Computing.** Lookup tables (tax bands, colour ramps, game difficulty curves) store a few points and draw straight lines between them; the line through two points is how a program fills the gaps.

Computer-science connection: lines without division

Geometry code prefers standard form, written as $F(x, y) = Ax + By - C$, because it never divides and treats vertical lines like any other. Through (x_1, y_1) and (x_2, y_2) you can take $A = y_2 - y_1$, $B = x_1 - x_2$, $C = Ax_1 + By_1$ — no slope needed. Better still, the **sign** of $F(x, y)$ tells you which side of the line a point is on ($F = 0$ means on it). That single sign test underlies polygon clipping, triangle rasterisation on GPUs, and "which side of the wall is the player on?" checks.

PYTHON

```python
from fractions import Fraction
from math import gcd

def line_through(p, q):
    """Integer (A, B, C) with Ax + By = C through integer points p != q."""
    (x1, y1), (x2, y2) = p, q
    A = y2 - y1
    B = x1 - x2
    C = A * x1 + B * y1
    g = gcd(gcd(A, B), C)                # reduce to lowest terms
    A, B, C = A // g, B // g, C // g
    if A < 0 or (A == 0 and B < 0):      # normalise the sign
        A, B, C = -A, -B, -C
    return A, B, C

def slope_intercept(A, B, C):
    """Convert Ax + By = C to (m, b) with y = mx + b; None if vertical."""
    if B == 0:
        return None
    return Fraction(-A, B), Fraction(C, B)

print(line_through((1, 2), (3, 8)), slope_intercept(*line_through((1, 2), (3, 8))))
print(line_through((5, -1), (5, 7)), slope_intercept(*line_through((5, -1), (5, 7))))
print(line_through((0, 3), (4, 0)), slope_intercept(*line_through((0, 3), (4, 0))))
```

```
(3, -1, 1) (Fraction(3, 1), Fraction(-1, 1))
(1, 0, 5) None
(3, 4, 12) (Fraction(-3, 4), Fraction(3, 1))
```



Horizontal and vertical are easy to swap by accident

$y = 3$ is a **horizontal** line (every point has height 3, any x). $x = 3$ is a **vertical** line. Horizontal lines have slope 0; vertical lines have **no** slope — "undefined", not zero, and not infinity in the sense of a number you can compute with. If a program computes `(y2 - y1) / (x2 - x1)` without checking, vertical lines crash it or produce `inf`.

US curriculum

Slope and slope-intercept form are core **Algebra I** (grades 8–9; slope first appears in **grade 8**). Point-slope and standard form are taught in Algebra I and revisited in **Geometry** (coordinate proofs, parallel and perpendicular lines) and **Algebra II**.

10.5.8. Exercises

1. Find the slope of the line through $(-1, 4)$ and $(3, -4)$, and write the line in all three forms.
2. Convert $4x - 3y = 12$ to slope-intercept form and state both intercepts.
3. Which form would you choose to describe the line through $(2, -1)$ and $(2, 6)$, and why?
4. A taxi charges a fixed fee plus a rate per km. A 3 km trip costs \$8.50 and an 8 km trip costs \$16. Write cost y as a function of distance x and interpret the slope and intercept.

▼ **Answers**

1. $m = \frac{-4-4}{3-(-1)} = \frac{-8}{4} = -2$. Point-slope: $y - 4 = -2(x + 1)$. Expand: $y - 4 = -2x - 2$, so slope-intercept $y = -2x + 2$. Standard: $2x + y = 2$. Check $(3, -4)$: $6 - 4 = 2$. Correct.
2. $-3y = -4x + 12$, divide by -3: $y = \frac{4}{3}x - 4$. x-intercept $(3, 0)$, y-intercept $(0, -4)$.
3. Both points have $x = 2$, so the line is vertical: $x = 2$. Only standard form ($1x + 0y = 2$) can express it; slope-intercept cannot.
4. Points $(3, 8.50)$ and $(8, 16)$. $m = \frac{16-8.5}{8-3} = \frac{7.5}{5} = 1.5$. Point-slope: $y - 8.5 = 1.5(x - 3)$, so $y = 1.5x + 4$. Slope: \$1.50 per km. Intercept: a \$4 fixed fee (the cost of a 0 km trip).

10.6. Parallel and perpendicular lines

10.6.1. The problem

A city planner wants a new street through a given point running **parallel** to an existing avenue, and a footpath from a building that meets the avenue at a **right angle** (the shortest route). Given the avenue's equation, what are the slopes of these new lines?

10.6.2. Intuition

Parallel lines never meet, so neither can be steeper than the other: they have **equal slopes** (and different intercepts — equal slope and equal intercept would be the same line). Vertical lines are all parallel to each other.

Perpendicular lines are subtler. Take the run–rise triangle of a line with slope $m_1 = \frac{1}{3}$: run 3, rise 1. Now rotate the whole picture by 90° anticlockwise. The run of 3 becomes a vertical rise of 3; the rise of 1 becomes a horizontal step of 1 **to the left**. The rotated line therefore has slope $\frac{3}{-1} = -3$. The slope flipped upside down **and** changed sign.

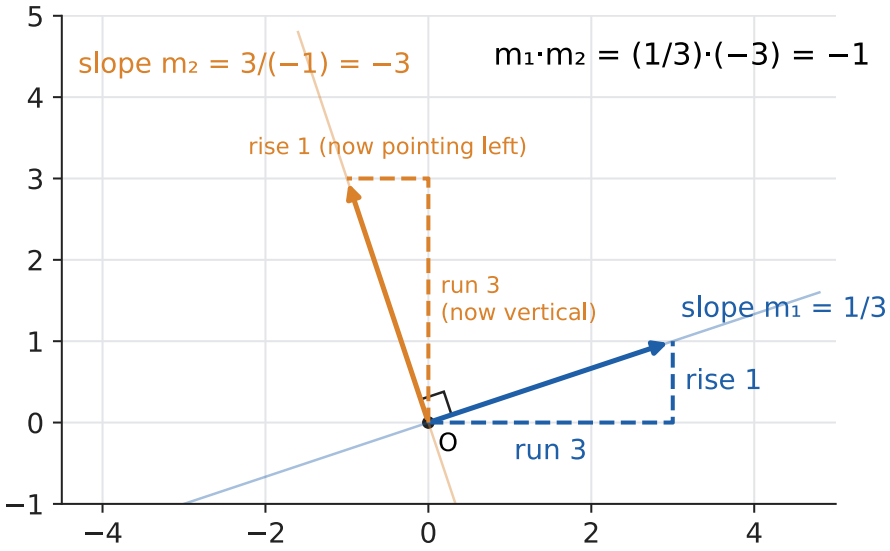


Figure 76. Rotating a line's run–rise triangle by 90° swaps rise and run and introduces a minus sign.

10.6.3. Why it works: two derivations

By rotation. A line with slope $m_1 = \frac{a}{b}$ has direction "run b, rise a", i.e. moves from the origin to (b, a). Rotating a point 90° anticlockwise about the origin sends (x, y) to $(-y, x)$ (we justify this rule in Section 10.9). So the perpendicular direction is $(-a, b)$, with slope $m_2 = \frac{b}{-a} = -\frac{b}{a}$. Hence

$$m_1 m_2 = \frac{a}{b} \cdot \left(-\frac{b}{a}\right) = -1.$$

By Pythagoras. Shift both lines so they pass through the origin (that changes neither slope nor angle): $y = m_1 x$ and $y = m_2 x$. They contain the points $A = (1, m_1)$ and $B = (1, m_2)$. The angle at O is a right angle exactly when $OA^2 + OB^2 = AB^2$ (Pythagoras and its converse, Chapter 9):

$$\begin{aligned}
(1 + m_1^2) + (1 + m_2^2) &= (1 - 1)^2 + (m_1 - m_2)^2 && \text{(distance formula three times)} \\
2 + m_1^2 + m_2^2 &= m_1^2 - 2m_1 m_2 + m_2^2 && \text{(expand the right side)} \\
2 &= -2m_1 m_2 && \text{(cancel } m_1^2 \text{ and } m_2^2\text{)} \\
m_1 m_2 &= -1 && \text{(divide by } -2\text{)}
\end{aligned}$$

Every step is reversible, so the condition is both necessary and sufficient.

Parallel and perpendicular slopes

Two non-vertical lines with slopes m_1, m_2 are

- **parallel** exactly when $m_1 = m_2$;
- **perpendicular** exactly when $m_1 m_2 = -1$, i.e. $m_2 = -\dfrac{1}{m_1}$: the **negative reciprocal**.

The **reciprocal** of $\frac{a}{b}$ is $\frac{b}{a}$ (flip it); the negative reciprocal flips and changes sign. Special case: horizontal ($m = 0$) and vertical lines are perpendicular, but the product rule cannot express this since a vertical slope does not exist.

10.6.4. The method

Example 61. Procedure — parallel or perpendicular line through a point

Input: a line L and a point $P = (x_1, y_1)$.

Steps:

1. Find the slope m of L (rearrange to $y = mx + b$, or use $m = -A/B$ from standard form).
2. Parallel: use slope m. Perpendicular: use $-1/m$ (if $m = 0$, the perpendicular is the vertical line $x = x_1$; if L is vertical, the perpendicular is horizontal, $y = y_1$).
3. Write point-slope form through P and simplify.
4. Check: P satisfies the new equation, and the slope product is -1 (perpendicular) or the slopes match (parallel).

Output: the equation of the required line.

10.6.5. Worked examples

Perpendicular through a point. Find the line through $(4, 1)$ perpendicular to $y = \frac{2}{3}x + 1$.

$$\begin{aligned} m_\perp &= -\frac{1}{2/3} = -\frac{3}{2} && \text{(negative reciprocal)} \\ y - 1 &= -\tfrac{3}{2}(x - 4) && \text{(point-slope)} \\ y - 1 &= -\tfrac{3}{2}x + 6 && \text{(expand: } -\tfrac{3}{2} \cdot -4 = 6\text{)} \\ y &= -\tfrac{3}{2}x + 7 && \text{(add 1)} \end{aligned}$$

Check: $-\frac{3}{2}(4) + 7 = 1$; $\frac{2}{3} \cdot (-\frac{3}{2}) = -1$. In standard form, $3x + 2y = 14$.

Perpendicular bisector. The **perpendicular bisector** of a segment is the line through its midpoint at right angles to it. Every point on it is equally far from both endpoints, which makes it the answer to "where are all the places equally close to A and B?". For $A(1, 2)$ and $B(7, 6)$:

$$
\begin{aligned}
M &= \left(\tfrac{1+7}{2}, \tfrac{2+6}{2}\right) = (4, 4) && \text{(midpoint)} \\
m_{AB} &= \frac{6-2}{7-1} = \frac{4}{6} = \frac{2}{3} && \text{(slope of the segment)} \\
m_{\perp} &= -\tfrac{3}{2} && \text{(negative reciprocal)} \\
y - 4 &= -\tfrac{3}{2}(x - 4) \implies y = -\tfrac{3}{2}x + 10 && \text{(point-slope, then simplify)}
\end{aligned}
$$

Check: $(0, 10)$ is on this line. Its squared distance to A is $1 + 64 = 65$; to B, $49 + 16 = 65$. Equal, as promised.

10.6.6. Distance from a point to a line

The shortest path from a point to a line is the perpendicular one. Doing the perpendicular construction in general (find the perpendicular through the point, intersect it with the line, apply the distance formula) gives a compact formula: the distance from (x_0, y_0) to the line $Ax + By = C$ is

$$
d = \frac{|Ax_0 + By_0 - C|}{\sqrt{A^2 + B^2}}.
$$

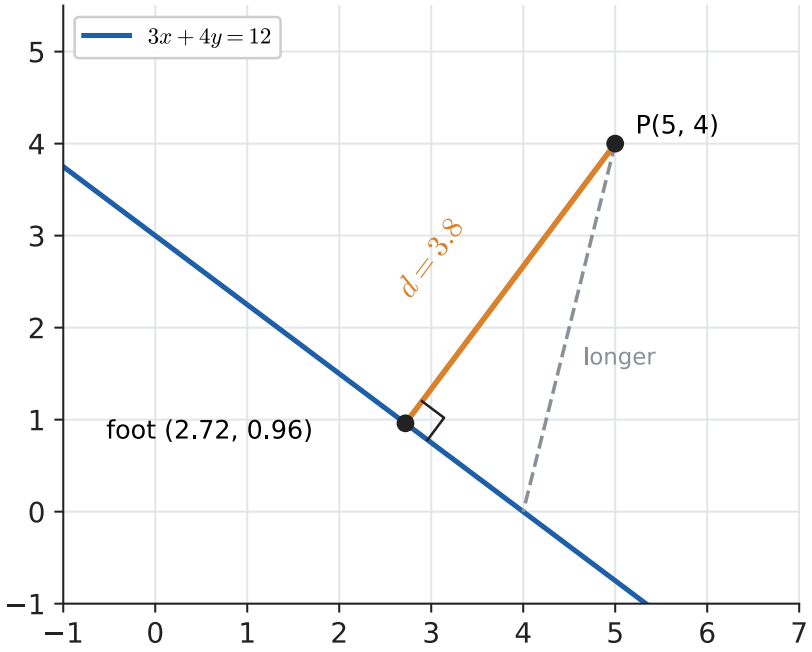


Figure 77. The shortest route from a point to a line meets the line at a right angle; any other route (dashed) is longer.

Example: from $(5, 4)$ to $3x + 4y = 12$, $d = \frac{|15+16-12|}{\sqrt{9+16}} = \frac{19}{5} = 3.8$. To confirm by the long route: the perpendicular through $(5, 4)$ has direction $(3, 4)$, so its points are $(5 + 3t, 4 + 4t)$. Substituting into the line, $3(5 + 3t) + 4(4 + 4t) = 12 \Rightarrow 31 + 25t = 12 \Rightarrow t = -0.76$, giving foot $(2.72, 0.96)$. The distance is $|t| \cdot \sqrt{3^2 + 4^2} = 0.76 \times 5 = 3.8$. The two routes agree. Notice the numerator is just $|F(x_0, y_0)|$, the side-of-line test from the previous CS sidebar: its sign says *which* side, and divided by $\sqrt{A^2 + B^2}$ its size says *how far*.

10.6.7. Where you'd use this

- **Voronoi diagrams.** Given several cell towers, which tower is nearest to each location? The boundary between two towers' regions is the perpendicular bisector of the segment joining them. Voronoi diagrams built from these bisectors appear in mobile networks, epidemiology (John Snow's 1854 cholera map is essentially one), and nearest-neighbour search.
- **Circumcentre.** The perpendicular bisectors of a triangle's three sides meet at one point, equidistant from all three vertices — the centre of the circle through them.
- **Shortest connection.** The footpath in our opening problem runs along the perpendicular from the building to the avenue; its length is the point-to-line distance.

10.6.8. Exercises

1. Find the line through $(2, 1)$ parallel to $y = -x + 7$.
2. Find the line through the origin perpendicular to $2x + 5y = 10$.
3. Find the perpendicular bisector of the segment from $(-1, 3)$ to $(5, -1)$, and verify one point on it is equidistant from both endpoints.
4. Find the distance from $(1, -2)$ to the line $5x - 12y = 3$.
5. Are the lines $3x - 4y = 7$ and $8x + 6y = 1$ parallel, perpendicular, or neither?

▼ **Answers**

1. Slope -1. $y - 1 = -(x - 2)$, so $y = -x + 3$.
2. Given slope $-A/B = -\frac{2}{5}$; perpendicular slope $\frac{5}{2}$. Through the origin: $y = \frac{5}{2}x$.
3. Midpoint $(2, 1)$. Segment slope $\frac{-1-3}{5-(-1)} = -\frac{2}{3}$; perpendicular slope $\frac{3}{2}$. $y - 1 = \frac{3}{2}(x - 2)$, so $y = \frac{3}{2}x - 2$. Check with $(0, -2)$: squared distance to $(-1, 3)$ is $1 + 25 = 26$; to $(5, -1)$ is $25 + 1 = 26$. Equal.
4. $d = \frac{|5(1) - 12(-2) - 3|}{\sqrt{25 + 144}} = \frac{|5 + 24 - 3|}{13} = \frac{26}{13} = 2$.
5. Slopes: $-A/B = \frac{3}{4}$ and $-\frac{8}{6} = -\frac{4}{3}$. Product $\frac{3}{4} \cdot (-\frac{4}{3}) = -1$: perpendicular.

10.7. Circles

10.7.1. The problem

A phone mast has a range of 5 km and stands at $(3, -2)$ on a map grid. Which grid points get signal? Equally often you are handed the problem the other way round: an equation like $x^2 + y^2 - 6x + 4y - 12 = 0$ comes out of some calculation, and you need to recognise it as a circle and find its centre and radius.

10.7.2. Intuition and derivation

A **circle** is the set of all points at a fixed distance (the radius) from a fixed point (the centre). That definition *is* the distance formula. A point (x, y) lies on the circle with centre (h, k) and radius $r > 0$ exactly when

$$\sqrt{(x-h)^2+(y-k)^2} = r \qquad \text{(distance from } (x,y) \text{ to the centre is } r\text{)}$$
$$(x-h)^2+(y-k)^2 = r^2 \qquad \text{(square both sides; both are non-negative, so nothing is lost)}$$

So the mast's coverage boundary is $(x-3)^2+(y+2)^2 = 25$, and the points with signal are those with $(x-3)^2+(y+2)^2 \le 25$. Note the sign flip: centre $k=-2$ appears as $(y-(-2))^2 = (y+2)^2$.



Equation of a circle

The **standard form** (or **centre–radius form**) of a circle with centre (h,k) and radius r is $(x-h)^2+(y-k)^2 = r^2$. The circle centred at the origin is $x^2+y^2=r^2$; with $r=1$ it is the **unit circle**, the star of Chapter 13. Expanding standard form gives the **general form** $x^2+y^2+Dx+Ey+F=0$.

10.7.3. The method: completing the square

Expanding $(x-3)^2+(y+2)^2=25$ gives $x^2-6x+9+y^2+4y+4=25$, i.e. $x^2+y^2-6x+4y-12=0$. To go **backwards** — from the general form to centre and radius — you complete the square in x and in y separately, exactly as in Chapter 8.

Example 62. Procedure — centre and radius from the general form

Input: an equation with x^2 and y^2 terms having **equal** coefficients and no xy term.

Steps:

1. If the x^2 and y^2 coefficients are not 1, divide the whole equation by that coefficient.
2. Group the x terms, group the y terms, and move the constant to the right.
3. For the x group x^2+Dx, add $(D/2)^2$ to **both** sides; it becomes $(x+D/2)^2$. Do the same for y.
4. Read off: centre $(h,k)=(-D/2,-E/2)$, and the right-hand side is r^2.
5. If the right-hand side is positive, r is its square root. If it is zero, the "circle" is the single point (h,k). If it is negative, there are no points at all.

Output: centre (h,k) and radius r, or a note that the equation is degenerate.

10.7.4. Worked example

$x^2+y^2-6x+4y-12=0$.

$$(x^2-6x\quad)+(y^2+4y\quad)=12 \qquad \text{(group, move constant)}$$
$$(x^2-6x+9)+(y^2+4y+4)=12+9+4 \qquad \text{(add } (\tfrac{-6}{2})^2=9 \text{ and } (\tfrac{4}{2})^2=4 \text{ to both sides)}$$
$$(x-3)^2+(y+2)^2=25 \qquad \text{(factor the perfect squares)}$$

Centre $(3,-2)$, radius $\sqrt{25}=5$: the phone mast again.

A version with a leading coefficient: $2x^2 + 2y^2 - 4x + 12y - 30 = 0$.

$$\begin{aligned} x^2 + y^2 - 2x + 6y - 15 &= 0 && \text{(divide by 2)} \\ (x^2 - 2x + 1) + (y^2 + 6y + 9) &= 15 + 1 + 9 && \text{(complete both squares)} \\ (x-1)^2 + (y+3)^2 &= 25 && \text{(centre } (1, -3)\text{, radius 5)} \end{aligned}$$

```python
import math

def circle_from_general(D, E, F):
    """x^2 + y^2 + Dx + Ey + F = 0  ->  (h, k, r), or None if no real circle."""
    h, k = -D / 2, -E / 2
    r_squared = h*h + k*k - F
    if r_squared < 0:
        return None                      # no points at all
    return h, k, math.sqrt(r_squared)

print(circle_from_general(-6, 4, -12))  # x^2+y^2-6x+4y-12=0
print(circle_from_general(2, 2, 10))
```

```
(3.0, -2.0, 5.0)
None
```

The second call is $x^2 + y^2 + 2x + 2y + 10 = 0$, i.e. $(x+1)^2 + (y+1)^2 = -8$: a sum of squares cannot be negative, so no point satisfies it.

10.7.5. Lines meeting circles, and tangents

Where does the line $y = x + 1$ meet the circle $x^2 + y^2 = 25$? Substitute the line into the circle — the substitution method of Chapter 6: *Try it:* Line meets circle, discriminant .

$$\begin{aligned} x^2 + (x+1)^2 &= 25 && \text{(replace } y\text{)} \\ 2x^2 + 2x + 1 &= 25 && \text{(expand)} \\ x^2 + x - 12 &= 0 && \text{(subtract 25, divide by 2)} \\ (x+4)(x-3) &= 0 && \text{(factor)} \end{aligned}$$

So $x = 3, y = 4$ or $x = -4, y = -3$: two crossing points. In general, the resulting quadratic's **discriminant** (Chapter 8) tells the story: positive means the line cuts the circle twice (a **secant**), zero means it touches once (a **tangent**), negative means it misses.

A **tangent** to a circle is perpendicular to the radius at the point of contact (Chapter 9). That turns "find the tangent" into a slope problem. Tangent to $x^2 + y^2 = 25$ at $(3, 4)$: the radius has slope $\frac{4}{3}$, so the tangent has slope $-\frac{3}{4}$, and $y - 4 = -\frac{3}{4}(x - 3)$, which simplifies to $3x + 4y = 25$. *Try it:* See the tangent touch .

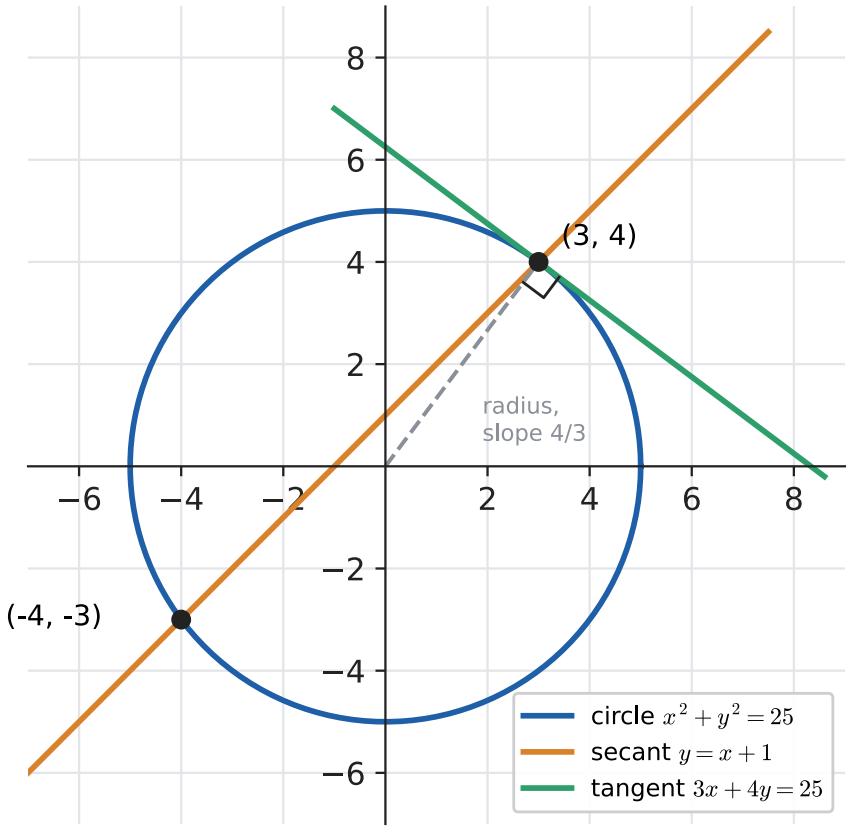


Figure 78. The secant $y = x + 1$ cuts the circle $x^2 + y^2 = 25$ at two points; the tangent at $(3, 4)$ touches it once, at right angles to the radius.



Not every $x^2 + y^2$ equation is a circle

- $x^2 + 4y^2 = 4$ has unequal coefficients: it is an **ellipse**, not a circle.
- $x^2 - y^2 = 4$ has opposite signs: a **hyperbola**.
- $(x - 1)^2 + (y + 2)^2 = 9$ has radius 3, not 9. The right-hand side is r^2.
- Forgetting to add the completing-the-square constants to **both** sides is the commonest error of all.

10.7.6. Where you'd use this

- **Delivery and coverage zones.** A pizza shop at map position $(2, 1)$ (in km) delivers within 3 km. Do you, at $(4, 3)$, qualify? $(4 - 2)^2 + (3 - 1)^2 = 8 \le 9 = 3^2$: yes, just. Phone masts, Wi-Fi access points, fire-station response areas and "stores near me" searches all run this inequality.
- **Positioning by trilateration.** If you know you are 5 km from beacon $A(0, 0)$ and 5 km from beacon $B(6, 0)$, you are on both circles $x^2 + y^2 = 25$ and $(x - 6)^2 + y^2 = 25$. Subtracting the equations kills the squares: $x^2 - (x - 6)^2 = 0$, so $12x - 36 = 0$ and $x = 3$; then $y^2 = 16$, so $y = \pm 4$. A third beacon picks between $(3, 4)$ and $(3, -4)$. GPS does the same with spheres in 3D (plus one more satellite to fix the receiver's clock error), and indoor positioning does it with Wi-Fi or Bluetooth signal strengths.

- **Engineering and graphics.** Gear teeth, cams, roundabouts and railway curves are laid out from circle equations; a CNC machine cutting an arc and a font renderer drawing a rounded corner both need centre and radius. Tangent lines matter wherever a straight piece must join a curve smoothly: a road leaving a roundabout, a belt leaving a pulley.

> **US curriculum**
>
> The circle equation, derived from the distance formula and recovered by completing the square, is a named Common Core standard in high-school **Geometry** (Expressing Geometric Properties with Equations). It is revisited with the other conics in **Algebra II** or **Precalculus**.

10.7.7. Exercises

1. Write the equation of the circle with centre $(-1, 4)$ and radius 3.
2. Find the centre and radius of $x^2 + y^2 + 8x - 2y + 8 = 0$.
3. Is the point $(4, 4)$ inside, on, or outside the circle $x^2 + y^2 - 6x + 4y - 12 = 0$?
4. Find the tangent to $x^2 + y^2 = 13$ at $(2, -3)$.
5. Find where the line $y = 7 - x$ meets the circle $x^2 + y^2 = 25$. *Try it:* Check your answer .

▼ **Answers**

1. $(x + 1)^2 + (y - 4)^2 = 9$.
2. $(x^2 + 8x + 16) + (y^2 - 2y + 1) = -8 + 16 + 1 = 9$, so $(x + 4)^2 + (y - 1)^2 = 9$: centre $(-4, 1)$, radius 3.
3. In standard form the circle is $(x - 3)^2 + (y + 2)^2 = 25$. At $(4, 4)$: $1^2 + 6^2 = 37 > 25$, so **outside**.
4. Radius slope $\frac{-3}{2}$; tangent slope $\frac{2}{3}$. $y + 3 = \frac{2}{3}(x - 2) \to 3y + 9 = 2x - 4 \to 2x - 3y = 13$. Check: $4 + 9 = 13$. Correct.
5. $x^2 + (7 - x)^2 = 25 \to 2x^2 - 14x + 49 = 25 \to x^2 - 7x + 12 = 0 \to (x - 3)(x - 4) = 0$. Points $(3, 4)$ and $(4, 3)$.

10.8. Conic sections

10.8.1. The problem

Why are satellite dishes and car headlights shaped the way they are? Why do planets trace ovals and some comets swing past the Sun once and never return? And what do all of those have to do with the circle you just met?

10.8.2. Intuition: slicing a cone

Take a double cone (two ice-cream cones joined at their tips) and slice it with a plane. Depending on the tilt of the slice you get exactly four kinds of curve, the **conic sections**: *Try it:* Slice the cone yourself .

- slice straight across: a **circle**;
- tilt a little: a closed oval, an **ellipse**;
- tilt until the plane is parallel to the side of the cone: an open U-shape, a **parabola**;
- tilt further still, cutting both halves of the cone: a two-piece curve, a **hyperbola**.

In coordinates, every conic is the graph of a **second-degree** equation $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$. Lines are degree 1; conics are the next step up.



Conic section

A **conic section** (**conic**) is a curve obtained by intersecting a plane with a double cone: circle, ellipse, parabola or hyperbola. Each (except the circle) can also be defined by a **focus** (a fixed point; plural **foci**) and a **directrix** (a fixed line). The **vertex** is the point where a parabola turns; the **axis of symmetry** is the line it is mirror-symmetric about.

10.8.3. Parabola: equal distances to a point and a line

A **parabola** is the set of points equally far from a focus and a directrix. Put the focus at $(0, p)$ and the directrix at $y = -p$. For a point (x, y), the distance to the focus is $\sqrt{x^2 + (y-p)^2}$ and the distance to the line is $|y + p|$. Setting them equal:

$$\begin{aligned} \sqrt{x^2 + (y-p)^2} &= |y+p| && \text{(definition)} \\ x^2 + (y-p)^2 &= (y+p)^2 && \text{(square both sides)} \\ x^2 + y^2 - 2py + p^2 &= y^2 + 2py + p^2 && \text{(expand)} \\ x^2 &= 4py && \text{(cancel } y^2 \text{ and } p^2\text{, add } 2py\text{)} \end{aligned}$$

So $y = \frac{x^2}{4p}$: the familiar parabola of Chapter 8, now with geometric meaning attached to its coefficient. For $y = \frac{1}{4}x^2$, $4p = 4$, so the focus is $(0, 1)$ and the directrix is $y = -1$. If $p < 0$ the parabola opens downward. *Try it:* Focus and directrix of y = x²/4 .

The reflective property explains the dish: every ray arriving parallel to the axis bounces off a parabolic mirror straight through the focus. Put the receiver at the focus and it collects all the signal; put a bulb there and the headlight sends out a parallel beam. *Try it:* Watch the rays reflect .

10.8.4. Ellipse and hyperbola: sums and differences

An **ellipse** is the set of points whose distances to two foci F_1, F_2 **add up** to a constant $2a$. (Pin a loop of string at two points and trace around with a pencil.) With foci $(\pm c, 0)$, the algebra — the same square-twice routine as the parabola, just longer — gives

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \qquad b^2 = a^2 - c^2.$$

a is the **semi-major axis** (half the longest width), b the **semi-minor axis**. When $c = 0$ the foci merge, $a = b$, and you get a circle.

A **hyperbola** is the set of points whose distances to two foci **differ** by a constant $2a$ (in absolute value). With foci $(\pm c, 0)$:

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1, \qquad b^2 = c^2 - a^2,$$

with **asymptotes** $y = \pm \frac{b}{a} x$ — lines the branches approach but never reach.

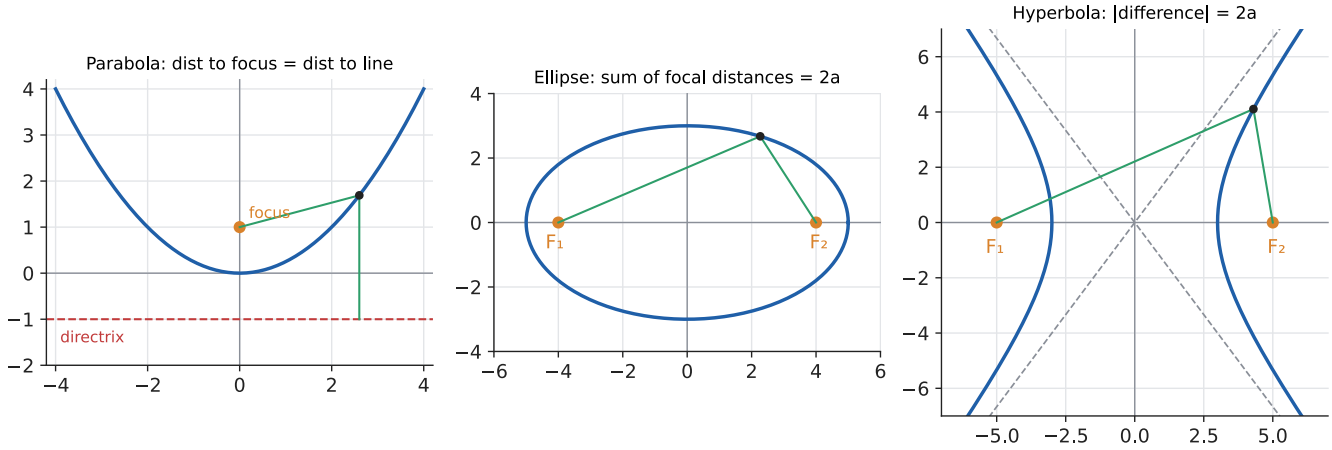


Figure 79. The three conics with their focal properties: parabola y = x²/4, ellipse x²/25 + y²/9 = 1 (foci ±4), hyperbola x²/9 - y²/16 = 1 (foci ±5).



Eccentricity

The **eccentricity** e of a conic is the ratio (distance to focus) / (distance to directrix), which is the same for every point on it. $e = 0$: circle. $0 < e < 1$: ellipse (for the standard ellipse, $e = c/a$). $e = 1$: parabola. $e > 1$: hyperbola. One number classifies them all.

Worked example: identify each conic and its key features.

- $\frac{x^2}{25} + \frac{y^2}{9} = 1$: ellipse with $a = 5, b = 3, c = \sqrt{25 - 9} = 4$. Foci $(\pm 4, 0)$, eccentricity $\frac{4}{5} = 0.8$. Check with the point $(0, 3)$: its distances to the foci are each $\sqrt{16 + 9} = 5$, summing to $10 = 2a$. *Try it:* This ellipse from its focus .
- $\frac{x^2}{9} - \frac{y^2}{16} = 1$: hyperbola, $a = 3, b = 4, c = \sqrt{9 + 16} = 5$. Foci $(\pm 5, 0)$, asymptotes $y = \pm \frac{4}{3} x$. Check with the vertex $(3, 0)$: distances 8 and 2, difference $6 = 2a$. *Try it:* This hyperbola from its focus .
- $x^2 = -12y$: parabola, $4p = -12$, $p = -3$. Opens downward; focus $(0, -3)$, directrix $y = 3$. *Try it:* This parabola from its focus .

10.8.5. Where you'd use this

- **Orbits.** Kepler's first law: planets move on ellipses with the Sun at one focus. Earth's orbit has $e \approx 0.017$ (nearly a circle); Halley's comet has $e \approx 0.97$. The focus is not at the centre, so the distance to the Sun varies: with semi-major axis $a \approx 149.6$ million km, the closest approach is

$a(1-e) \approx 147.1$ million km (early January) and the farthest $a(1+e) \approx 152.1$ million km (early July). Seasons come from the tilt of Earth's axis, not this 3% difference. An object moving faster than escape velocity follows a hyperbola and leaves forever.

- **Reflectors.** Parabolic dishes, telescope mirrors, solar cookers. Elliptical "whispering galleries" carry a whisper from one focus to the other; lithotripsy machines focus shock waves on a kidney stone the same way.
- **Positioning.** If two radio stations transmit simultaneously and you measure the *difference* in arrival times, you know the difference of your distances to them: you are on a hyperbola with the stations as foci. Two such hyperbolas pin you down. This was the LORAN navigation system, and the same time-difference idea underlies GPS and microphone-array sound location.
- **Projectiles.** Ignoring air resistance, a thrown ball follows a parabola.

Computer-science connection: classifying a conic

Graphics, CAD and computer-vision code meet the general second-degree equation $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$ constantly (for example, when fitting an ellipse to detected edge points). The quantity $B^2 - 4AC$ — a discriminant again — classifies a non-degenerate conic: negative means ellipse (a circle if also $B = 0$ and $A = C$), zero means parabola, positive means hyperbola. The Bxy term means the conic is **rotated**; removing it is an eigenvector problem you will be able to solve after Volume 2, Chapter 28 .



Apollonius and Kepler

Apollonius of Perga (c. 200 BCE) wrote eight books on conics and gave us the names *ellipse* ("falling short"), *parabola* ("thrown alongside", i.e. exactly matching) and *hyperbola* ("thrown beyond", i.e. exceeding) — originally describing how an area compared with another in his constructions. The same roots give "ellipsis" (words left out), "parable" and "hyperbole" (exaggeration). For 1800 years conics were a purely geometric curiosity, until Kepler discovered in 1609 that the planets move on ellipses.



Focus and directrix playground

Place a focus and a directrix, then drag a slider for eccentricity e. The app plots every point whose focus-distance is e times its directrix-distance. Watch the ellipse stretch as $e \to 1$, snap into a parabola at $e = 1$, and split into a hyperbola beyond. A second mode slices a 3D cone with a tiltable plane.

Try it: Open the focus–directrix playground · Slide the eccentricity

US curriculum

The parabola from focus and directrix is in the Common Core **Geometry** standards. Full treatment of ellipses and hyperbolas is typically in **Algebra II** or **Precalculus** (sometimes as a "conic sections" unit), and the sum/difference-of-distances definitions may be optional depending on the district.

10.8.6. Exercises

1. Identify each conic and give its foci: (a) $\frac{x^2}{16} + \frac{y^2}{9} = 1$; (b) $y = \frac{x^2}{8}$; (c) $\frac{x^2}{4} - \frac{y^2}{5} = 1$.
2. A satellite dish is a parabola in cross-section, 60 cm across and 9 cm deep. How far from the vertex should the receiver go? *Try it:* Check your answer .
3. Find the equation of the ellipse with foci $(\pm 3, 0)$ and major axis of length 10.

▼ **Answers**

1. (a) Ellipse, $a = 4, b = 3, c = \sqrt{16 - 9} = \sqrt{7}$; foci $(\pm\sqrt{7}, 0)$. (b) Parabola, $4p = 8$, $p = 2$; focus $(0, 2)$, directrix $y = -2$. (c) Hyperbola, $a = 2, b = \sqrt{5}, c = \sqrt{4 + 5} = 3$; foci $(\pm 3, 0)$, asymptotes $y = \pm\frac{\sqrt{5}}{2}x$.
2. Model as $x^2 = 4py$ with the vertex at the origin. The rim is at $(30, 9)$: $900 = 4p \cdot 9 = 36p$, so $p = 25$. The receiver goes 25 cm from the vertex along the axis.
3. $2a = 10 \Rightarrow a = 5$; $c = 3$; $b^2 = a^2 - c^2 = 16$. $\frac{x^2}{25} + \frac{y^2}{16} = 1$.

10.9. Transformations as coordinate rules

10.9.1. The problem

A game character is drawn as a list of vertex coordinates. The player presses "turn left": every vertex must rotate 90° about the character's centre. Then the camera zooms, the level scrolls, a mirror-image enemy appears. Each of these is a rule that takes a point's coordinates and produces new ones. What are the rules?

10.9.2. Intuition

In Chapter 9 you met congruence (same shape and size) and similarity (same shape, scaled). Transformations are the *moves* that produce congruent or similar copies. On the coordinate plane each move is a function from points to points, written $(x, y) \mapsto (x', y')$ — read "(x, y) maps to (x', y')". The original is the **pre-image**; the result is the **image**, conventionally labelled with primes ($A \mapsto A'$).



Transformation, image, isometry

A **transformation** of the plane is a rule that sends every point to a new point. The **image** of A is written A' ("A prime"). An **isometry** or **rigid motion** preserves all distances (and hence angles and areas): translations, reflections and rotations are isometries, and they produce **congruent** figures. A **dilation** scales all distances by the same factor and produces **similar** figures.

10.9.3. The four basic rules

Translation (slide) by a across and b up: $(x, y) \mapsto (x + a, y + b)$. Every point moves by the same vector (a, b).

Reflection (mirror):

- in the x-axis: $(x, y) \mapsto (x, -y)$ — the height flips sign;
- in the y-axis: $(x, y) \mapsto (-x, y)$;
- in the line $y = x$: $(x, y) \mapsto (y, x)$ — the coordinates swap;
- in the origin (point reflection): $(x, y) \mapsto (-x, -y)$.

Rotation about the origin, anticlockwise:

- by 90°: $(x, y) \mapsto (-y, x)$;
- by 180°: $(x, y) \mapsto (-x, -y)$ (the same as the point reflection);
- by 270° (= 90° clockwise): $(x, y) \mapsto (y, -x)$.

Dilation (scaling) about the origin with scale factor k: $(x, y) \mapsto (kx, ky)$. $|k| > 1$ enlarges, $0 < |k| < 1$ shrinks; a negative k also rotates by 180°.

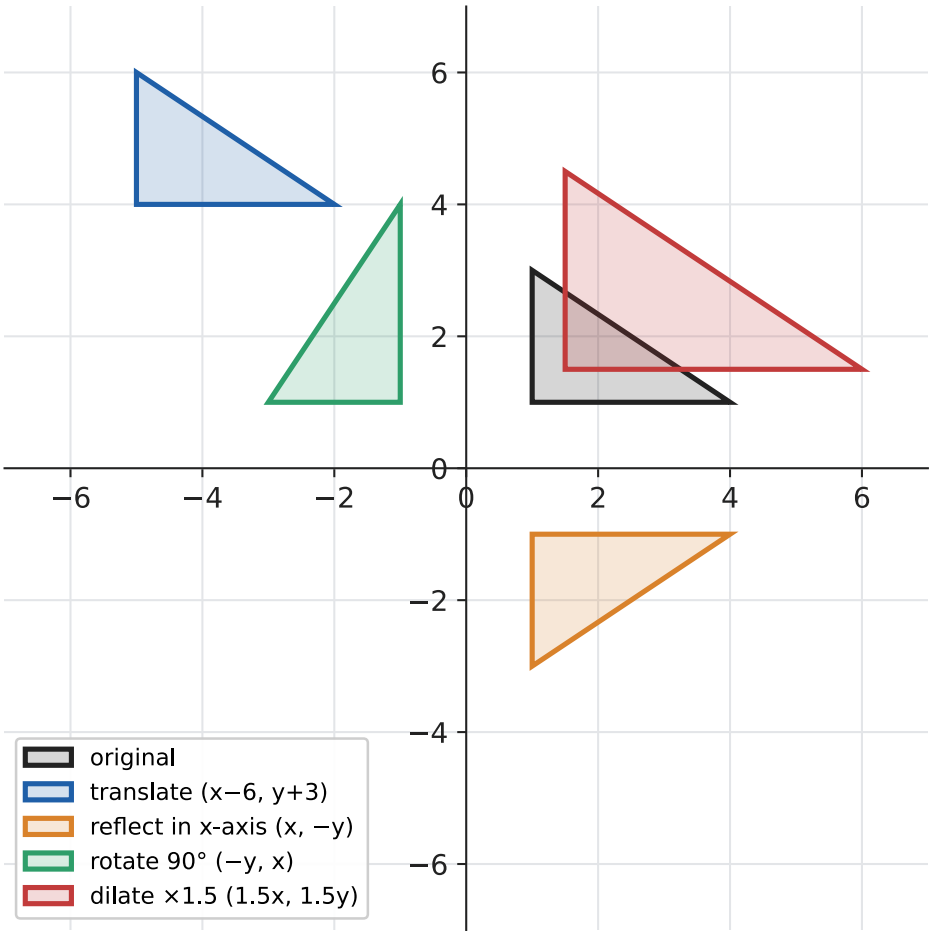


Figure 80. The four basic transformations applied to the triangle (1,1), (4,1), (1,3).

10.9.4. Why the rotation rule works

Think of (x, y) as "x steps along the x-direction $(1, 0)$, plus y steps along the y-direction $(0, 1)$". A 90° anticlockwise turn sends the direction $(1, 0)$ (east) to $(0, 1)$ (north), and sends $(0, 1)$ (north) to $(-1, 0)$ (west). Rotating the whole picture rotates the step directions and keeps the step counts:

$$\begin{aligned} (x, y) &= x\,(1, 0) + y\,(0, 1) && \text{(steps east and north)} \\ &\mapsto x\,(0, 1) + y\,(-1, 0) && \text{(east becomes north, north becomes west)} \\ &= (-y, x) \end{aligned}$$

The same argument works for any angle θ: east goes to $(\cos\theta, \sin\theta)$ and north to $(-\sin\theta, \cos\theta)$, so

$$(x, y) \mapsto (x\cos\theta - y\sin\theta,\ x\sin\theta + y\cos\theta).$$

You will meet **cos** and **sin** in Chapter 13; check that $\theta = 90°$ ($\cos = 0$, $\sin = 1$) gives back $(-y, x)$. The idea "a transformation is determined by where it sends the two basic directions" is exactly what a **matrix** records. The 90° rotation is the matrix $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$, and composing transformations becomes multiplying matrices — the whole subject of Volume 2, Chapter 27 .

10.9.5. The method: transforming about any point

The rotation and dilation rules above use the origin as their fixed centre. To rotate or scale about another point, move that point to the origin, transform, and move back.

Example 63. Procedure — rotate (or dilate) about a centre C

Input: a point $P = (x, y)$, a centre $C = (c_1, c_2)$, and a rotation (or dilation) rule about the origin.

Steps:

1. Translate so the centre is at the origin: $(u, v) = (x - c_1, y - c_2)$.
2. Apply the origin-centred rule to (u, v), giving (u', v').
3. Translate back: $P' = (u' + c_1, v' + c_2)$.
4. Check (rotation): P and P' are the same distance from C.

Output: the image P'.

10.9.6. Worked examples

A triangle, rotated. Rotate $A(1, 1)$, $B(4, 1)$, $C(1, 3)$ by 90° anticlockwise about the origin with $(x, y) \mapsto (-y, x)$: $A' = (-1, 1)$, $B' = (-1, 4)$, $C' = (-3, 1)$. Check a side: $AB = 3$, and $A'B' = \sqrt{0^2 + 3^2} = 3$. Distances preserved, as an isometry must. *Try it:* Triangle rotated 90° ; Rotate P(5,3) about C(2,1) .

About another centre. Rotate $P(5, 3)$ by 90° anticlockwise about $C(2, 1)$.

$$\begin{aligned} (u, v) &= (5 - 2,\ 3 - 1) = (3, 2) && \text{(move C to the origin)} \\ (u', v') &= (-2, 3) && \text{(apply } (u, v) \mapsto (-v, u)\text{)} \\ P' &= (-2 + 2,\ 3 + 1) = (0, 4) && \text{(move back)} \end{aligned}$$

Check: $CP = \sqrt{3^2 + 2^2} = \sqrt{13}$ and $CP' = \sqrt{(-2)^2 + 3^2} = \sqrt{13}$. Equal.

10.9.7. Order matters: composition

Doing one transformation and then another is called **composition**. Sometimes the order is irrelevant; often it is not.

```python
import math

def translate(p, a, b):  return (p[0] + a, p[1] + b)
def reflect_x(p):        return (p[0], -p[1])
def reflect_y(p):        return (-p[0], p[1])
def reflect_diag(p):     return (p[1], p[0])          # in the line y = x
def rotate90(p):         return (-p[1], p[0])         # 90° anticlockwise about O
def dilate(p, k):        return (k * p[0], k * p[1])  # centre O, factor k

def rotate(p, theta):
    """Rotate p anticlockwise about the origin by theta radians."""
    c, s = math.cos(theta), math.sin(theta)
    return (c * p[0] - s * p[1], s * p[0] + c * p[1])

tri = [(1, 1), (4, 1), (1, 3)]
print([rotate90(p) for p in tri])
print([tuple(round(v, 12) for v in rotate(p, math.pi / 2)) for p in tri])
print([reflect_x(translate(p, 2, 0)) for p in tri])
print([translate(reflect_x(p), 2, 0) for p in tri])
print([rotate90(reflect_x(p)) for p in tri], [reflect_x(rotate90(p)) for p in tri])
```

```
[(-1, 1), (-1, 4), (-3, 1)]
[(-1.0, 1.0), (-1.0, 4.0), (-3.0, 1.0)]
[(3, -1), (6, -1), (3, -3)]
[(3, -1), (6, -1), (3, -3)]
[(1, 1), (1, 4), (3, 1)] [(-1, -1), (-1, -4), (-3, -1)]
```

Try it: Order matters: reflect, rotate .

A horizontal slide and a reflection in the x-axis **commute** (the order does not matter), because one changes only x and the other only y. But "reflect in the x-axis, then rotate 90°" sends $(x, y) \to (x, -y) \to (y, x)$ — a reflection in $y = x$ — while the opposite order sends $(x, y) \to (-y, x) \to (-y, -x)$, a reflection in $y = -x$. Different answers. The floating-point `rotate` agrees with the exact `rotate90` only after rounding: $\cos(\pi/2)$ in floating point is about 6×10^{-17}, not 0.



Rotation direction and the sign of the angle

Mathematics measures positive angles **anticlockwise**. "Rotate 90°" with no direction means anticlockwise. The rules $(-y, x)$ (90° anticlockwise) and $(y, -x)$ (90° clockwise) are easy to swap; test with $(1, 0)$: anticlockwise must send it to $(0, 1)$, straight up. On a screen with y pointing down (see Section 10.11), the same formula **looks** clockwise.

10.9.8. Where you'd use this

- **Signs and crafts.** Lettering stuck on the inside of a shop window must read correctly from the street, so the sign-maker cuts it reflected in a vertical line: $(x, y) \mapsto (-x, y)$. Iron-on T-shirt transfers are printed mirrored for the same reason. *Try it:* Enlarge about a centre .

- **Resizing about a centre.** Enlarging a logo by 150% about its centre $(10, 5)$ (in cm) sends a corner at $(12, 8)$ to $(10 + 1.5 \cdot 2,\ 5 + 1.5 \cdot 3) = (13, 9.5)$: translate the centre to the origin, dilate, translate back, exactly the procedure above. Embroidery machines, laser cutters and vinyl plotters apply these rules to every point of a design.
- **Science.** Chemists classify molecules and crystals by the rotations and reflections that leave them unchanged (their **symmetry group**). A molecule and its mirror image can behave very differently in the body; many drugs are sold as one "hand" only for that reason.
- **Machine learning.** Image classifiers are trained on flipped, rotated and slightly scaled copies of each photo (**data augmentation**), because a cat reflected in a vertical line is still a cat. Each copy is these coordinate rules applied to every pixel position.

Computer-science connection: the graphics pipeline

Every 2D and 3D renderer pushes vertex coordinates through a chain of transformations: model space → world space (place the object) → camera space (where the viewer is) → screen space (pixels). Rotations, reflections and dilations are **linear** (they keep the origin fixed and can be written as matrices); translations are not. The standard trick, **homogeneous coordinates**, writes (x, y) as $(x, y, 1)$ so that a translation also becomes a matrix multiplication, and the whole pipeline collapses into a single matrix — one multiply per vertex. CSS `transform: rotate(…) translate(…)` and SVG `transform="matrix(a b c d e f)"` expose exactly this; and since composition does not commute, `rotate` then `translate` differs from `translate` then `rotate`.



Transformation sandbox

Draw a triangle, then stack transformations (translate, reflect, rotate, dilate) in a list you can reorder. The app shows each intermediate image and the final rule $(x, y) \mapsto (x', y')$. Swap two items and see whether the result changes. A matrix panel shows the 2×2 (or 3×3 homogeneous) matrix of the composite.

Try it: Transformation sandbox

US curriculum

Common Core puts transformations at the centre of **grade 8** (describe the effect of dilations, translations, rotations and reflections using coordinates) and of high-school **Geometry**, where congruence is *defined* as "related by a sequence of rigid motions" and similarity as "rigid motions plus a dilation". Matrix representations appear in **Precalculus** or later.

10.9.9. Exercises

1. Find the image of $(2, -5)$ under: (a) reflection in $y = x$; (b) rotation by 180° about the origin; (c) the translation $(x, y) \mapsto (x - 3, y + 4)$; (d) dilation by factor 3 about the origin.

2. Rotate $(4, 1)$ by 90° anticlockwise about $(1, 1)$.
3. Show that reflecting in the x-axis and then in the y-axis is the same as rotating by 180°.
4. Identify the transformation $(x, y) \mapsto (y, -x)$.
5. A dilation with factor 3 is applied to a triangle of area 4. What are the new perimeter ratio and area?

▼ **Answers**

1. (a) $(-5, 2)$. (b) $(-2, 5)$. (c) $(-1, -1)$. (d) $(6, -15)$.
2. Shift: $(4 - 1, 1 - 1) = (3, 0)$. Rotate: $(0, 3)$. Shift back: $(1, 4)$. Check: both are 3 units from $(1, 1)$.
3. $(x, y) \mapsto (x, -y) \mapsto (-x, -y)$, which is the 180° rotation rule.
4. Test with $(1, 0) \mapsto (0, -1)$: east goes to south, a quarter turn clockwise. It is rotation by 90° clockwise (equivalently 270° anticlockwise) about the origin.
5. Every length is multiplied by 3, so the perimeter triples. Area scales by $3^2 = 9$: the new area is 36. (Similar figures, Chapter 9.)

10.10. Three dimensions

10.10.1. Intuition and vocabulary

Add a third axis, the **z-axis**, perpendicular to both the x- and y-axes. A point in space now has three coordinates (x, y, z). The axes are arranged by the **right-hand rule**: curl the fingers of your right hand from the positive x-axis towards the positive y-axis, and your thumb points along the positive z-axis. The three **coordinate planes** (the xy-, yz- and xz-planes) divide space into eight **octants**.



Three-dimensional coordinates

3D Cartesian coordinates (x, y, z) locate a point in space by signed distances along three mutually perpendicular axes. The system is **right-handed** when x, y, z follow the right-hand rule (the standard convention in mathematics and physics). Space, thought of as all such triples, is written $\mathbb{R}^3$ ("R three"); the plane is $\mathbb{R}^2$.

10.10.2. Why the distance formula extends

From $P_1 = (x_1, y_1, z_1)$ to $P_2 = (x_2, y_2, z_2)$, first travel in the floor (the xy-directions) to the point $Q = (x_2, y_2, z_1)$ directly below or above P_2; then go straight up. Pythagoras twice:

$$P_1Q^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \qquad \text{(2D distance within the floor)}$$

$$P_1P_2^2 = P_1Q^2 + QP_2^2 \qquad (QP_2 \text{ is vertical, so the angle at } Q \text{ is right})$$

$$= (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

The pattern continues to any number of dimensions: square each coordinate difference, add, take the root. In Volume 2, Chapter 26 this becomes the **length** (norm) of a vector, and it works just as well in 300 dimensions as in 3 — which is how machine-learning embeddings measure similarity.

The midpoint is again the coordinate-wise average, and the **sphere** with centre (h, k, l) and radius r is $(x-h)^2 + (y-k)^2 + (z-l)^2 = r^2$. A single linear equation $ax + by + cz = d$ describes a **plane**, not a line — one equation removes one degree of freedom from three.

10.10.3. Worked example

The length of the main diagonal of a box 2 by 3 by 6 (from $(0,0,0)$ to $(2,3,6)$):

$$d = \sqrt{2^2 + 3^2 + 6^2} = \sqrt{4 + 9 + 36} = \sqrt{49} = 7.$$

Is the point $(3,2,0)$ inside the sphere of radius 4 centred at $(1,0,-2)$? Squared distance $2^2 + 2^2 + 2^2 = 12 < 16 = 4^2$: inside.

Where you'd use this. Will a 2.4 m curtain pole fit in a car boot measuring 1.0 m by 1.1 m by 0.5 m? The longest straight line inside the box is the space diagonal, $\sqrt{1.0^2 + 1.1^2 + 0.5^2} = \sqrt{2.46} \approx 1.57$ m, so no. In medical imaging, a CT scan is a 3D grid of voxels whose spacing is often different along each axis, say 0.5 mm, 0.5 mm and 1.0 mm. To measure a tumour from voxel $(10, 20, 5)$ to voxel $(40, 60, 15)$, convert to millimetres first (differences $30 \times 0.5 = 15$, $40 \times 0.5 = 20$, $10 \times 1.0 = 10$) and then apply the formula: $\sqrt{225 + 400 + 100} = \sqrt{725} \approx 26.9$ mm. Applying the formula to raw voxel indices would give the wrong answer, a classic bug. GPS receivers, drones, 3D printers (every line of G-code is a (x, y, z) move) and robot arms all work in 3D coordinates.

Computer-science connection: 3D graphics and colour spaces

3D engines store every vertex as (x, y, z). Be warned that conventions vary: OpenGL uses a right-handed world with y up; DirectX traditionally used a left-handed system; Blender has z up; Unity is left-handed with y up. Converting a model between tools is often "swap two axes and negate one".

Coordinates need not be spatial. A colour is a point (r, g, b) in a cube; the distance formula measures how different two colours are (crudely — perceptual colour spaces like CIELAB were designed so that this distance better matches human vision). Lerping between colour points gives gradients.

10.10.4. Exercises

1. Find the distance between $(2, -1, 3)$ and $(5, 3, 15)$.
2. Find the midpoint of $(1, 2, 3)$ and $(5, -2, 7)$.
3. Write the equation of the sphere with centre $(0, 0, 0)$ passing through $(1, -2, 2)$.

▼ **Answers**

1. Differences $3, 4, 12$: $d = \sqrt{9 + 16 + 144} = \sqrt{169} = 13$.
2. $(3, 0, 5)$.
3. $r^2 = 1 + 4 + 4 = 9$: $x^2 + y^2 + z^2 = 9$ (radius 3).

10.11. Screen coordinates and pixel grids

10.11.1. The problem

You have plotted $y = x^2$ on paper. Now draw it in a browser canvas, an image file, or a terminal. The first attempt comes out upside down. Why?

10.11.2. Intuition

Displays are scanned like text: left to right, top to bottom. So almost every screen, image format and windowing system puts the origin at the **top-left corner**, with x increasing to the right and y increasing **downward**. Pixel coordinates are also integers, and they number **cells**, not points: pixel $(0, 0)$ is the little square in the corner.

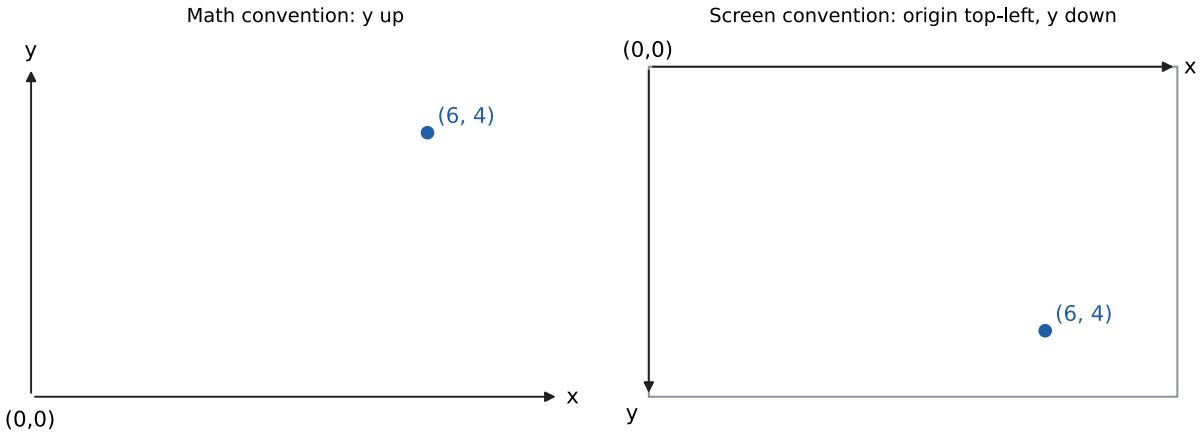


Figure 81. The same coordinates (6, 4) in mathematical and screen conventions.



Screen coordinates and pixels

Screen (or **device**, **pixel**) **coordinates** place the origin at the top-left, with x (column) rightward and y (row) downward. A **pixel** ("picture element") is one cell of the grid. **World coordinates** are the mathematical coordinates of the scene being drawn. A **viewport transform** converts world to screen coordinates.

10.11.3. The method: world to screen

To draw a mathematical scene centred on (c_x, c_y) at s pixels per unit in a canvas W pixels wide and H high, combine three transformations from Section 10.9: translate the world centre to the origin, dilate by s (and reflect in the x-axis to flip y), then translate the origin to the canvas centre: *Try it:* Try the world-to-screen map .

$$\text{col} = \frac{W}{2} + s\,(x - c_x), \qquad \text{row} = \frac{H}{2} - s\,(y - c_y).$$

The single minus sign in the row formula is the whole difference between "math" and "screen".

```python
def world_to_screen(x, y, width, height, scale, cx=0.0, cy=0.0):
    """Map math coords (y up, centred on (cx, cy)) to pixel coords (y down)."""
    col = width / 2 + (x - cx) * scale
    row = height / 2 - (y - cy) * scale        # the minus sign flips y
    return round(col), round(row)

print(world_to_screen(0, 0, 800, 600, 50))
print(world_to_screen(3, 2, 800, 600, 50))
print(world_to_screen(-8, -6, 800, 600, 50))
```

```
(400, 300)
(550, 200)
(0, 600)
```

The world point $(3, 2)$, up and right of centre, lands **above** the centre row 300, at row 200, as it should.



Flipped y flips everything that depends on orientation

In screen coordinates a line that rises on screen has a **negative** slope numerically; an anticlockwise rotation formula **appears** clockwise; the "left-hand side" test from a line's sign flips; and polygon vertex order that was anticlockwise becomes clockwise (which can turn a shoelace area negative, or make a renderer cull the faces you wanted to see). Decide on one convention per layer of your code and convert exactly once, at the boundary.

US curriculum

Screen coordinates are not part of the standard math sequence; they appear in **AP Computer Science** and in middle- and high-school coding courses (Scratch's stage, for example, is math-style with $(0, 0)$ in the centre and y up, while most text-based languages use the screen convention).

10.12. Drawing a line on a pixel grid: Bresenham's algorithm

10.12.1. The problem

A line on paper is infinitely thin and continuous. A screen can only light whole pixels. Given two pixel endpoints, which pixels should be lit so the result looks like a straight line — no gaps, no clumps — and can we do it fast, ideally using only integer addition?

10.12.2. Intuition

Consider a line whose slope is between 0 and 1 (gently rising, more across than up), from (x_0, y_0) to (x_1, y_1) with integer endpoints. Because it is flat-ish, we want **exactly one pixel per column**. Moving one column to the right, the line rises by $m < 1$, so the next pixel is either straight across, $(x + 1, y)$,

or diagonally up, $(x+1, y+1)$. Choose whichever is closer to the true line. The tie-breaker is the **midpoint** between the two candidates, $(x+1, y+\frac{1}{2})$: if the true line passes above it, go up; otherwise go across.

The naive way computes $y = y_0 + m(x - x_0)$ for each column and rounds. It works, but it uses floating point and a multiplication per pixel — expensive on 1960s plotters (where Jack Bresenham invented the method, at IBM in 1962) and still wasteful today. Bresenham's insight is to track the **sign** of an integer error term that changes by a fixed amount each step.

10.12.3. Why it works

Let $\Delta x = x_1 - x_0 > 0$ and $\Delta y = y_1 - y_0$, with $0 \le \Delta y \le \Delta x$. Write the line in the standard form of Section 10.5, scaled to integers:

$$F(x, y) = \Delta y\,(x - x_0) - \Delta x\,(y - y_0).$$

$F = 0$ on the line; $F > 0$ for points **below** it (smaller y makes $-\Delta x\,(y - y_0)$ larger). Evaluate at the midpoint and double to clear the $\frac{1}{2}$: define the **decision variable** $D = 2F(x+1, y+\frac{1}{2})$.

- If $D > 0$, the midpoint is below the line, so the line is closer to the upper pixel: choose $(x+1, y+1)$.
- Otherwise choose $(x+1, y)$.

How does D change from one column to the next? F is linear, so moving one unit in x adds Δy to F (that is, $2\Delta y$ to D), and moving one unit up in y subtracts Δx ($2\Delta x$ from D). At the start, $F(x_0, y_0) = 0$, so

$$D_{\text{start}} = 2F(x_0 + 1, y_0 + \tfrac{1}{2}) = 2\left(\Delta y - \tfrac{1}{2}\Delta x\right) = 2\Delta y - \Delta x,$$

$$D \leftarrow D + 2\Delta y \qquad \text{(after stepping across)}$$

$$D \leftarrow D + 2\Delta y - 2\Delta x \qquad \text{(after stepping diagonally up)}$$

Only integer additions and comparisons. No multiplication inside the loop, no rounding, no floating point.

Example 64. Procedure — Bresenham's line algorithm (first octant)

Input: integer endpoints (x_0, y_0), (x_1, y_1) with $x_0 < x_1$ and $0 \le y_1 - y_0 \le x_1 - x_0$ (slope between 0 and 1).

Steps:

1. Set $\Delta x = x_1 - x_0$, $\Delta y = y_1 - y_0$, $D = 2\Delta y - \Delta x$, $y = y_0$.
2. For each x from x_0 to x_1 inclusive:
   a. Plot pixel (x, y).
   b. If $D > 0$: increase y by 1 and subtract $2\Delta x$ from D.
   c. Add $2\Delta y$ to D.

Output: the list of pixels, one per column, each within half a pixel of the true line.

Other directions: for steep lines (slope > 1) swap the roles of x and y; for leftward or downward lines step by -1 instead of $+1$. The general version below handles all eight octants at once.

10.12.4. Worked example

Draw the segment from $(1, 1)$ to $(8, 4)$. Here $\Delta x = 7$, $\Delta y = 3$, so $2\Delta y = 6$, $2\Delta x = 14$, and $D_{\text{start}} = 6 - 7 = -1$. *Try it:* Step through this example .

x	plot y	D on arrival	decision	true y at this x
1	1	−1	$D \le 0$: across; $D = -1 + 6 = 5$	1.000
2	1	5	$D > 0$: up; $D = 5 - 14 + 6 = -3$	1.429
3	2	−3	across; $D = 3$	1.857
4	2	3	up; $D = 3 - 14 + 6 = -5$	2.286
5	3	−5	across; $D = 1$	2.714
6	3	1	up; $D = 1 - 14 + 6 = -7$	3.143
7	4	−7	across; $D = -1$	3.571
8	4	−1	(endpoint)	4.000

Pixels: $(1,1), (2,1), (3,2), (4,2), (5,3), (6,3), (7,4), (8,4)$. Compare each plotted y with the true y in the last column: every one is the nearest integer. (At $x = 2$, for example, the true height is 1.429, closer to 1 than to 2, and the algorithm plotted 1.)

```python
def bresenham_octant1(x0, y0, x1, y1):
    """Pixels for a line with x0 < x1 and 0 <= slope <= 1."""
    dx, dy = x1 - x0, y1 - y0
    D = 2 * dy - dx                    # decision variable
    y = y0
    pixels = []
    for x in range(x0, x1 + 1):
        pixels.append((x, y))
        if D > 0:                      # true line is above the midpoint
            y += 1
            D -= 2 * dx
        D += 2 * dy
    return pixels

print(bresenham_octant1(1, 1, 8, 4))
```

```
[(1, 1), (2, 1), (3, 2), (4, 2), (5, 3), (6, 3), (7, 4), (8, 4)]
```

The production version handles every direction with one loop by keeping signed step directions s_x, s_y and a combined error term: *Try it:* Run the all-octant version .

```python
def bresenham(x0, y0, x1, y1):
    """Integer pixels on the segment (x0, y0)-(x1, y1); all octants."""
    pixels = []
    dx = abs(x1 - x0)
    dy = -abs(y1 - y0)                 # stored negative: simplifies the tests
    sx = 1 if x0 < x1 else -1          # step direction in x
    sy = 1 if y0 < y1 else -1          # step direction in y
    err = dx + dy                      # error term, scaled by 2 below
    while True:
        pixels.append((x0, y0))
        if x0 == x1 and y0 == y1:
            return pixels
        e2 = 2 * err
        if e2 >= dy:                   # time to step in x
            err += dy
            x0 += sx
        if e2 <= dx:                   # time to step in y
            err += dx
            y0 += sy

print(bresenham(0, 0, 10, 4))
print(bresenham(0, 0, 2, 5))
print(bresenham(3, 3, -2, 1))
```

```
[(0, 0), (1, 0), (2, 1), (3, 1), (4, 2), (5, 2), (6, 2), (7, 3), (8, 3), (9, 4), (10, 4)]
[(0, 0), (0, 1), (1, 2), (1, 3), (2, 4), (2, 5)]
[(3, 3), (2, 3), (1, 2), (0, 2), (-1, 1), (-2, 1)]
```

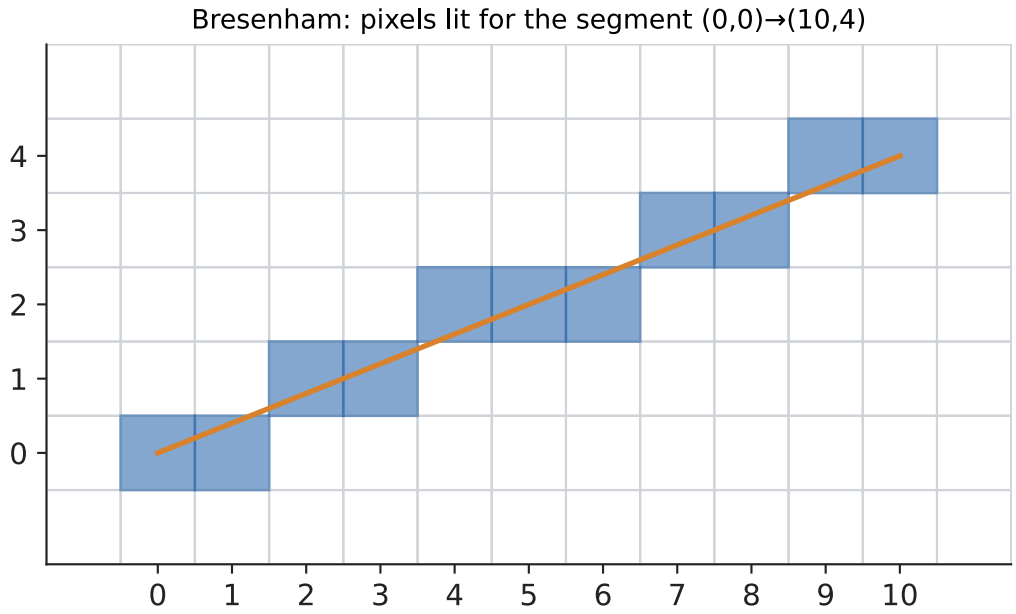


Figure 82. Bresenham's algorithm lighting pixels for the segment from (0, 0) to (10, 4).

10.12.5. Alternatives and choosing

- **DDA** (digital differential analyser): step x by 1 and add the floating-point slope m to y each step, rounding to plot. Simpler to explain; uses floating point; accumulated rounding error can drift on very long lines.
- **Bresenham / midpoint**: integer-only and exact. The classic choice for aliased (hard-edged) lines, and the same midpoint idea draws circles (the **midpoint circle algorithm**) using the circle equation's sign instead of the line's.
- **Anti-aliased lines** (Xiaolin Wu's algorithm): light two pixels per column with brightness proportional to closeness. Smoother-looking; use when visual quality matters more than crispness.

Today GPUs rasterise lines and triangles in hardware, but the decision-variable idea — evaluate a linear function's sign incrementally — is exactly how triangle rasterisers decide which pixels a triangle covers.



Ties are a choice, not a bug

When the true line passes exactly through a midpoint ($D = 0$), both candidate pixels are equally good. Different implementations break the tie differently: for the segment $(0, 0) \to (4, 2)$, the first-octant version above plots $(1, 0)$ and $(3, 1)$, while the all-octant version plots $(1, 1)$ and $(3, 2)$. Both are valid. But drawing $A \to B$ and $B \to A$ may then light slightly different pixels — which matters if you later erase a line by redrawing it.

Try it: See the tie case .



Pixel-by-pixel

Click two pixels on a zoomed-in grid. Step through Bresenham one column at a time, watching the midpoint dot, the true line, and the value of D in a side panel. Toggle to DDA and to anti-aliased mode and compare the results on near-horizontal, 45°, and steep lines.

Try it: Open the pixel-by-pixel lab

10.13. Bounding boxes, point-in-circle tests, and distance metrics

10.13.1. The problem

A game has 500 moving objects. Checking whether every pair of detailed outlines overlaps is far too slow ($500 \times 499/2 \approx 125{,}000$ pairs per frame, each expensive). How can coordinate geometry give **cheap** tests that rule out most pairs immediately?

10.13.2. Axis-aligned bounding boxes

Wrap each object in the smallest rectangle with sides parallel to the axes that contains it: its **axis-aligned bounding box** (AABB), described by $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ — the minimum and maximum of its vertices' coordinates. Two such boxes overlap exactly when their x-intervals overlap **and** their y-intervals overlap. Two intervals $[a_1, a_2]$ and $[b_1, b_2]$ overlap when each starts before the other ends: $a_1 \le b_2$ and $b_1 \le a_2$.

If the boxes do not overlap, the objects certainly do not; only if they do is the expensive exact test needed. This "cheap test first" pattern is called a **broad phase** followed by a **narrow phase**.

10.13.3. Point in circle, circle against circle

A point (x, y) is inside or on the circle with centre (h, k) and radius r when $(x-h)^2 + (y-k)^2 \le r^2$ — the circle equation from Section 10.7, turned into an inequality, with no square root. Two circles overlap when the distance between centres is at most the sum of the radii: $(h_1 - h_2)^2 + (k_1 - k_2)^2 \le (r_1 + r_2)^2$.

PYTHON

```python
def bounding_box(points):
    xs = [x for x, _ in points]
    ys = [y for _, y in points]
    return min(xs), min(ys), max(xs), max(ys)   # (xmin, ymin, xmax, ymax)

def boxes_overlap(a, b):
    """Axis-aligned boxes (xmin, ymin, xmax, ymax); touching counts."""
    return a[0] <= b[2] and b[0] <= a[2] and a[1] <= b[3] and b[1] <= a[3]

def in_circle(p, centre, r):
    dx, dy = p[0] - centre[0], p[1] - centre[1]
    return dx*dx + dy*dy <= r*r                # no square root needed

ship = bounding_box([(2, 1), (6, 3), (4, 5)])
rock = bounding_box([(5, 4), (8, 6), (7, 9)])
print(ship, rock, boxes_overlap(ship, rock))
print(in_circle((5, 6), (2, 2), 5), in_circle((6, 6), (2, 2), 5))
```

```
(2, 1, 6, 5) (5, 4, 8, 9) True
True False
```

The ship's box spans $x \in [2, 6]$, $y \in [1, 5]$; the rock's spans $x \in [5, 8]$, $y \in [4, 9]$. Both interval pairs overlap, so a narrow-phase test is needed. The point $(5, 6)$ is at squared distance $9 + 16 = 25$ from $(2, 2)$: exactly on the radius-5 circle, so it counts as inside. The point $(6, 6)$ is at $16 + 16 = 32 > 25$: outside.



A box test can say "maybe" but never a false "no"

Overlapping bounding boxes do **not** mean the objects touch: two thin diagonal sticks can have heavily overlapping boxes while being far apart. AABBs give false positives, never false negatives, which is exactly why they are only a first filter. Also decide deliberately whether touching edges count ($\leq$) or not ($<$); mixing the two between functions causes flickering "is it colliding?" bugs.

10.13.4. Euclidean versus Manhattan distance

In Manhattan, a taxi cannot drive diagonally through buildings: it moves along streets in a grid. From $A(1, 1)$ to $B(5, 4)$ it must cover 4 blocks east and 3 blocks north, 7 blocks in total, whatever route it takes. The straight-line (Euclidean) distance is $\sqrt{4^2 + 3^2} = 5$. *Try it:* Euclidean versus Manhattan distance .

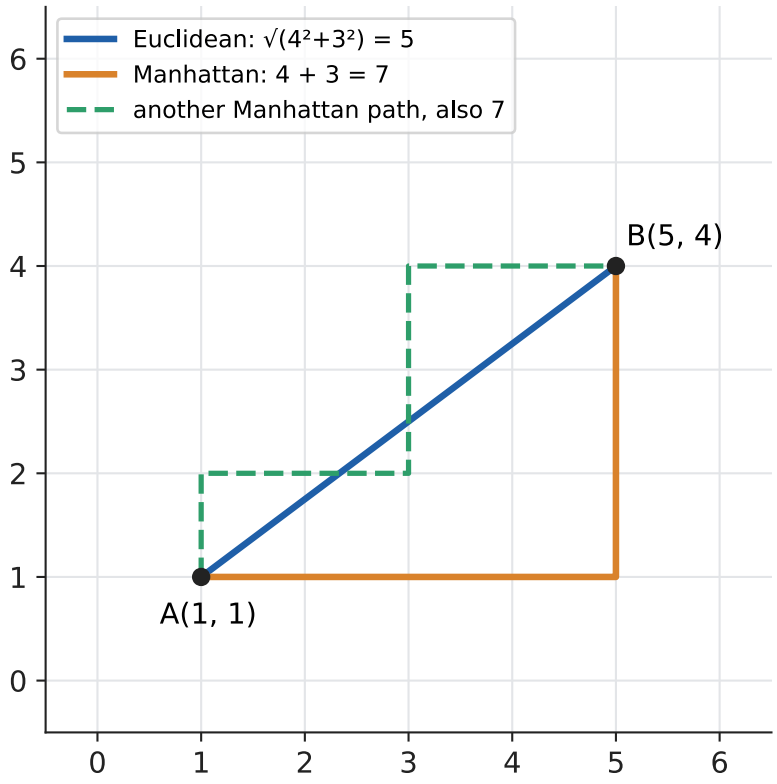


Figure 83. Euclidean distance (straight line) versus Manhattan distance (any monotone grid path) between the same two points.

Distance metrics

For points $p = (x_1, y_1)$ and $q = (x_2, y_2)$:

- **Euclidean** (L^2) distance: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ — the straight line.
- **Manhattan** (**taxicab**, L^1) distance: $|x_2 - x_1| + |y_2 - y_1|$ — grid moves only.
- **Chebyshev** (**chessboard**, L^∞) distance: $\max(|x_2 - x_1|, |y_2 - y_1|)$ — moves like a chess king, diagonals allowed at the same cost.

A **metric** is any distance function that is never negative, is zero only from a point to itself, is symmetric, and obeys the **triangle inequality** $d(p, r) \le d(p, q) + d(q, r)$ (a detour never helps). All three above are metrics.

PYTHON

```python
def euclidean(p, q):
    return ((p[0] - q[0])**2 + (p[1] - q[1])**2) ** 0.5

def manhattan(p, q):
    return abs(p[0] - q[0]) + abs(p[1] - q[1])

def chebyshev(p, q):
    return max(abs(p[0] - q[0]), abs(p[1] - q[1]))

A, B = (1, 1), (5, 4)
print(euclidean(A, B), manhattan(A, B), chebyshev(A, B))
```

Try it: Drag A and B .

```
5.0 7 4
```

Always $\textbf{Chebyshev} \le \textbf{Euclidean} \le \textbf{Manhattan}$: here $4 \le 5 \le 7$.

Choosing a metric. Use Euclidean when movement is free in any direction (physics, drones, k-nearest-neighbours on continuous features). Use Manhattan when movement is restricted to a grid (city routing, 4-directional game maps, circuit-board wiring) or when you want a cheaper, absolute-value-only computation that is less dominated by one large difference. Use Chebyshev when diagonal moves cost the same as straight ones (8-directional grids, chess kings, some warehouse cranes that move both axes at once).

Computer-science connection: heuristics for A* search

The A* path-finding algorithm (used in games, robotics and route planners; see Volume 3, Chapter 25) needs a **heuristic**: an estimate of the remaining distance to the goal that never **over**-estimates it. On a 4-directional grid, Manhattan distance is exactly right; Euclidean would underestimate (still correct, but slower). On an 8-directional grid with equal-cost diagonals, use Chebyshev; Manhattan would **overestimate** and could make A* return a non-shortest path. Choosing the metric that matches the movement rules is the difference between a correct, fast search and a subtly wrong one.

Try it: Watch Manhattan mislead A* .



Circles that are squares

A circle is "all points at distance r from a centre". Change the metric and the circle changes shape. In the Manhattan metric, the points at distance 1 from the origin satisfy $|x| + |y| = 1$: a square standing on its corner (a diamond). In the Chebyshev metric, $\max(|x|, |y|) = 1$ is an ordinary axis-aligned square. "Taxicab geometry" is a fully fledged non-Euclidean geometry in which many Euclidean theorems fail — for instance, two distinct points can have infinitely many midpoints.



Unit circles in three metrics

Pick a metric (Euclidean, Manhattan, Chebyshev, or a general L^p with a slider for p) and the app draws every point at distance 1 from a centre you can drag. Watch the circle morph from diamond ($p = 1$) through round ($p = 2$) to square ($p \to \infty$). A second panel runs A* on a grid with each metric as heuristic and counts the nodes explored.

Try it: When p < 1 breaks · Morph the unit circle, run A* · Morph the Lᵖ unit circle · Morph the unit circle with p

10.13.5. Exercises

1. A canvas is 640 by 480 pixels, with world origin at the centre and 20 pixels per unit. Where is the world point $(-4, 3)$ drawn? Which world point is drawn at pixel $(100, 400)$? *Try it:* Check your answer .
2. Use the first-octant Bresenham procedure to list the pixels from $(0, 0)$ to $(5, 2)$, showing D at each step. *Try it:* Check your answer .
3. Compute the Euclidean, Manhattan and Chebyshev distances between $(-2, 3)$ and $(4, -5)$.
4. Do the boxes $(0, 0, 4, 3)$ and $(5, 1, 7, 2)$ overlap? Do circles of radius 3 at $(0, 0)$ and radius 3.5 at $(4, 5)$ overlap?
5. A delivery robot moves in a warehouse along aisles parallel to the walls. Which metric should its route planner use as a heuristic, and why?

▼ **Answers**

1. col $= 320 + 20(-4) = 240$, row $= 240 - 20(3) = 180$: pixel $(240, 180)$. Inverting, $x = (\text{col} - 320)/20 = (100 - 320)/20 = -11$ and $y = (240 - \text{row})/20 = (240 - 400)/20 = -8$: world point $(-11, -8)$.
2. $\Delta x = 5$, $\Delta y = 2$, $D = 4 - 5 = -1$. x=0: plot (0,0), $D \le 0$, $D = -1 + 4 = 3$. x=1: plot (1,0), $D > 0$, $y = 1$, $D = 3 - 10 + 4 = -3$. x=2: plot (2,1), $D = 1$. x=3: plot (3,1), $D > 0$, $y = 2$, $D = 1 - 10 + 4 = -5$. x=4: plot (4,2), $D = -1$. x=5: plot (5,2). Pixels: $(0, 0), (1, 0), (2, 1), (3, 1), (4, 2), (5, 2)$.
3. $|\Delta x| = 6$, $|\Delta y| = 8$. Euclidean $\sqrt{36 + 64} = 10$; Manhattan 14; Chebyshev 8.
4. Boxes: the x-intervals $[0, 4]$ and $[5, 7]$ do not overlap ($5 > 4$), so no. Circles: squared centre distance $16 + 25 = 41$; $(3 + 3.5)^2 = 42.25$; $41 \le 42.25$, so they overlap (just).
5. Manhattan: the robot can only move parallel to the axes, so the Manhattan distance is the true length of an unobstructed route and never overestimates the real path.

10.14. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Geometry: triangles, circles, polygons, solids
- Conics, pixels and metrics
- Graphing calculator
- Matrix transformations
- Vectors playground

10.15. Mastery check

1. Find the distance between $(-1, -3)$ and $(7, 3)$, and their midpoint.
2. Find, in slope-intercept and standard form, the line through $(2, -3)$ perpendicular to $4x - 2y = 6$.
3. Find the centre and radius of $x^2 + y^2 - 10x + 6y + 18 = 0$.
4. Identify the conic $x^2 = -12y$ and give its focus and directrix.
5. The triangle $(0, 0), (2, 0), (0, 1)$ is rotated 90° anticlockwise about the origin and then translated by $(x, y) \mapsto (x + 3, y)$. Give the final vertices.
6. Find the distance between $(1, 1, 1)$ and $(3, -2, 7)$.
7. On an 800 by 600 canvas with the world origin at the centre and 50 pixels per unit, which pixel shows the world point $(2, -1)$?
8. A taxi drives from block $(1, 2)$ to block $(6, -4)$ on a street grid. How far does it drive, and how far is that as the crow flies?

Decision rule: All correct without hesitation → move on to Chapter 11. Stuck on 1–2 → reread the relevant section (Section 10.3, Section 10.6, Section 10.7, Section 10.8, Section 10.9, Section 10.11) and redo its exercises. Stuck on more → work through all the section exercises first, and revisit Pythagoras in Chapter 9 and completing the square in Chapter 8.

▼ **Mastery check answers**

1. $\Delta x = 8$, $\Delta y = 6$: $d = \sqrt{64 + 36} = 10$. Midpoint $\left(\frac{6}{2}, \frac{0}{2}\right) = (3, 0)$.
2. $4x - 2y = 6 \Rightarrow y = 2x - 3$, slope 2. Perpendicular slope $-\frac{1}{2}$.
   $y + 3 = -\frac{1}{2}(x - 2) = -\frac{1}{2}x + 1$, so $y = -\frac{1}{2}x - 2$. Standard form: multiply by 2,
   $2y = -x - 4$, so $x + 2y = -4$. Check $(2, -3)$: $2 - 6 = -4$. Correct.
3. $(x^2 - 10x + 25) + (y^2 + 6y + 9) = -18 + 25 + 9 = 16$, so $(x - 5)^2 + (y + 3)^2 = 16$: centre $(5, -3)$, radius 4.
4. Parabola of the form $x^2 = 4py$ with $4p = -12$, $p = -3$: opens downward, vertex at the origin, focus $(0, -3)$, directrix $y = 3$.
5. Rotate with $(x, y) \mapsto (-y, x)$: $(0, 0), (0, 2), (-1, 0)$. Translate: $(3, 0), (3, 2), (2, 0)$.
6. Differences $2, -3, 6$: $\sqrt{4 + 9 + 36} = \sqrt{49} = 7$.
7. col $= 400 + 50 \cdot 2 = 500$; row $= 300 - 50 \cdot (-1) = 350$. Pixel $(500, 350)$.
8. Manhattan: $|6 - 1| + |-4 - 2| = 5 + 6 = 11$ blocks. Euclidean: $\sqrt{25 + 36} = \sqrt{61} \approx 7.81$ blocks.

Part IV: Functions, Trigonometry, and Precalculus

Chapter 11. Functions and Their Graphs

A function is the single most important idea in this book. Everything you have done so far (evaluating expressions, solving linear and quadratic equations, plotting lines) was quietly about functions. This chapter makes the idea explicit: what a function is, the vocabulary that surrounds it (domain, codomain, range, composition, inverse), how to read and reshape graphs, and the main families you will meet again and again. Along the way you will see why a programmer's "function" is both the same thing and a slightly dangerous false friend.

Why it matters: almost every quantitative statement you will ever make is a function in disguise. Your phone bill as a function of data used, a drug's concentration as a function of time since the dose, a web page's load time as a function of server traffic, a neural network's prediction as a function of its input. Once you can see the function, you can graph it, invert it, combine it with others and ask where it breaks.

Where this chapter sits

You need: algebraic manipulation (Chapter 4), lines and slope (Chapter 5), quadratics and their graphs (Chapter 8), and the Cartesian plane (Chapter 10). Polynomials from Chapter 7 help with the section on rational functions.

This chapter unlocks: exponential and logarithmic functions (Chapter 12), the trigonometric functions (Chapter 13), sequences as functions on the integers (Chapter 14), limits and continuity (Volume 2, Chapter 16), the derivative (Volume 2, Chapter 17 , where composition becomes the chain rule), and linear maps (Volume 2, Chapter 27). Injective, surjective and bijective functions return in counting arguments (Volume 3, Chapter 24) and proofs (Volume 3, Chapter 22).

11.1. The idea of a function

11.1.1. The problem

A vending machine takes a button code and gives you a snack. Press B4 and you get crisps. Press B4 again tomorrow and you get crisps again. If B4 sometimes gave crisps and sometimes a chocolate bar, you would call the machine broken.

That reliability (same input, same output, every time) is the whole idea of a function. It is such a simple idea that it is easy to underrate. But once you have it, a huge amount of mathematics becomes one story: a line is a function, a parabola is a function, the area of a circle as its radius changes is a function, a sorted list is a function from positions to values, and a database index is a function from keys to rows.

The questions this section answers are: what exactly counts as a function, and how do we write one down?

11.1.2. Intuition: machines and arrows

There are two pictures worth keeping in your head.

The machine picture. A function is a box. You feed in an input; it returns exactly one output. You do not need to see inside the box; you only need to know that it is deterministic.

The mapping picture. Draw the set of allowed inputs on the left, the set of possible outputs on the right, and an arrow from each input to its output. The rule is: **every input has exactly one arrow leaving it.** An output may receive several arrows, or none at all. That asymmetry is the heart of the definition.

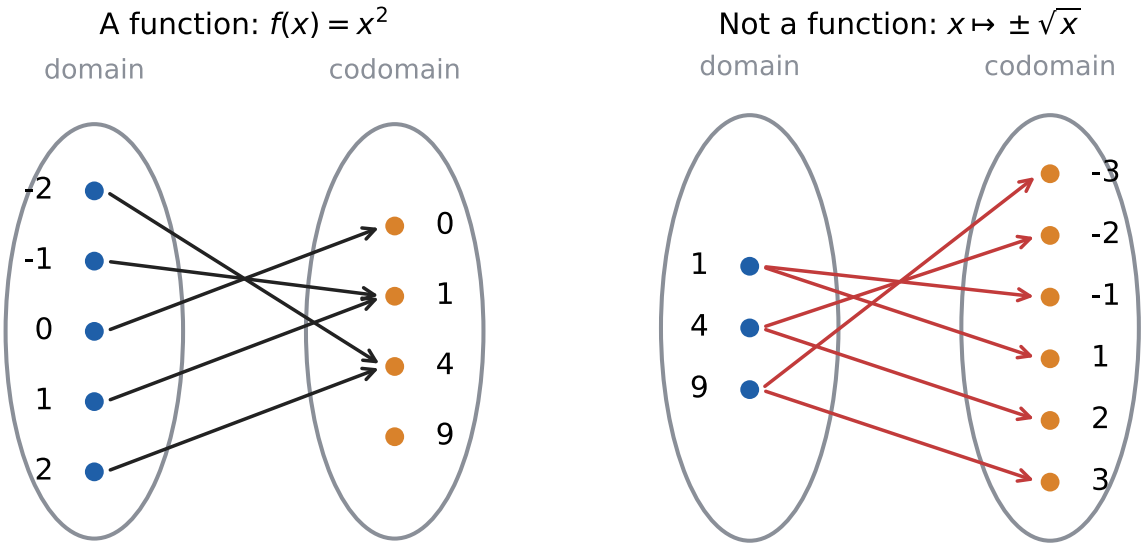


Figure 84. Left: squaring is a function (every input has exactly one arrow; 4 receives two arrows and 9 none, which is allowed). Right: "take the square roots" is not a function, because 1, 4 and 9 each send out two arrows.

Notice what the right-hand picture says. "Give me a number whose square is 9" has two honest answers, +3 and −3. A rule with two answers is a perfectly good **relation**, but it is not a function. This is exactly why mathematicians define $\sqrt{9}$ to mean the **non-negative** root only, 3: they wanted the square root to be a function.

11.1.3. Vocabulary and notation



Function

A **function** f from a set A to a set B is a rule that assigns to **each** element of A **exactly one** element of B.

We write $f : A \to B$, read "f maps A to B" or "f from A to B". The arrow is about the whole function; it says where inputs come from and where outputs live.



Input, output, argument, value, image

If f sends x to y, we write $f(x) = y$, read "f of x equals y".

- x is the **input**, also called the **argument** or **independent variable**.
- y is the **output**, also called the **value of f at x**, the **image of x**, or the **dependent variable** (it depends on x).

The parentheses in $f(x)$ mean "apply f to", **not** multiplication. $f(x)$ is one object, the output.



The "maps to" arrow and membership

$x \mapsto x^2$ (a barred arrow) reads "x maps to x^2". It describes what happens to a single input, and lets you talk about a function without naming it. Compare:

- $f : \mathbb{R} \to \mathbb{R}$ (plain arrow) says which sets are involved;
- $x \mapsto x^2$ (barred arrow) says what the rule does.

$x \in A$ reads "x is an element of A" (or "x is in A"). $\mathbb{R}$ is the set of all real numbers, $\mathbb{Z}$ the integers, $\mathbb{N}$ the natural numbers (see Chapter 1).

So "$f : \mathbb{R} \to \mathbb{R},\ x \mapsto x^2$" and "$f(x) = x^2$ for real x" say the same thing; the first is the style of university texts, the second of school texts.

11.1.4. Worked example: evaluating a function

Let $f(x) = 2x^2 - 3x + 1$. Find $f(3)$, $f(-1)$ and $f(a+1)$.

The rule is "whatever is in the parentheses, put it everywhere you see x". Use brackets when substituting; they prevent sign errors.

$$
\begin{aligned}
f(3) &= 2(3)^2 - 3(3) + 1 = 18 - 9 + 1 = 10 \\
f(-1) &= 2(-1)^2 - 3(-1) + 1 = 2 + 3 + 1 = 6 \\
f(a+1) &= 2(a+1)^2 - 3(a+1) + 1 && \text{(replace every } x \text{ by } (a+1)) \\
&= 2(a^2 + 2a + 1) - 3a - 3 + 1 && \text{(expand the square)} \\
&= 2a^2 + 4a + 2 - 3a - 2 \\
&= 2a^2 + a
\end{aligned}
$$

Check with $a = 2$: $f(3)$ should equal $2(2)^2 + 2 = 10$. It does.



$f(x)$ is not f times x

Two classic errors come from treating $f(\dots)$ like multiplication:

- $f(a+b) \neq f(a)+f(b)$ in general. With $f(x)=x^2$: $f(1+2)=9$ but $f(1)+f(2)=5$.
- $f(2x) \neq 2f(x)$ in general. With $f(x)=x^2$: $f(2x)=4x^2$, not $2x^2$.

Functions that **do** satisfy both rules are called **linear maps**. They are special, and they are the subject of Volume 2, Chapter 27 .

11.1.5. Four representations of one function

The same function can be shown in four ways, and fluency means moving between them freely:

Representation	**Example: the area of a square of side s**
Words	"The area is the side length squared."
Formula	$A(s)=s^2$, for $s \geq 0$
Table	$s=1,2,3,4$ gives $A=1,4,9,16$
Graph	the right half of the parabola $y=x^2$

A formula is compact and exact; a table is what a computer actually stores; a graph shows behaviour at a glance; words carry the meaning. None is "the" function. The function is the input-to-output assignment that all four describe.



Who invented "function"?

Leibniz used the word *functio* in the 1670s for quantities attached to a curve (such as its slope). Euler wrote $f(x)$ in 1734 and for a long time meant "a formula". The modern definition (any rule, formula or not, that assigns one output per input) emerged in the 19th century with Dirichlet, who gave a famous example that no single formula could plausibly express: $D(x)=1$ if x is rational and 0 if irrational. The final step, defining a function as a set of input–output pairs, came with set theory around 1900. Programmers recover the same idea when they say "a function is just a lookup table with a very efficient compression scheme."

US curriculum

The idea of a function, $f(x)$ notation, and the four representations are introduced in **Grade 8** (Common Core standard 8.F) and formalised in **Algebra I** (grades 8–9, standards F-IF). The $f : A \to B$ notation is rare at school and standard at university.

11.2. Domain, codomain and range

11.2.1. The problem

Your code has a function `sqrt(x)`. What happens if you call `sqrt(-4)`? In Python, `math.sqrt(-4)` raises `ValueError: math domain error`. The error message uses exactly the right mathematical word. Every function has a set of inputs it accepts, and asking for anything outside that set is meaningless (or, in the case of computers, an exception).

11.2.2. Vocabulary



Domain, codomain, range

For a function $f : A \to B$:

- The **domain** is A, the set of allowed inputs.
- The **codomain** is B, the set where outputs are declared to live (the "return type").
- The **range** (also called the **image**) is the set of outputs that actually occur: $\{f(x) : x \in A\}$. It is always a subset of the codomain, and may be smaller.

Example: $f : \mathbb{R} \to \mathbb{R},\ f(x) = x^2$. Domain $\mathbb{R}$, codomain $\mathbb{R}$, range $[0, \infty)$, since squares are never negative.

The codomain versus range distinction is exactly the distinction between a function's declared return type and the values it really returns. A Python function annotated `-> int` that only ever returns 0 or 1 has codomain `int` and range $\{0, 1\}$.



Interval notation and set-builder notation

Interval notation describes runs of real numbers. A square bracket includes the endpoint; a round bracket excludes it.

- $[2, 5]$ means $2 \le x \le 5$ (closed interval).
- $(2, 5)$ means $2 < x < 5$ (open interval).
- $[2, 5)$ means $2 \le x < 5$ (half-open).
- $[0, \infty)$ means $x \ge 0$. Infinity always takes a round bracket, because it is not a number you can reach.
- $\cup$ (union) joins pieces: $(-\infty, 2) \cup (2, \infty)$ is "every real except 2", also written $\mathbb{R} \setminus \{2\}$.

Set-builder notation $\{x \in \mathbb{R} : x \neq 2\}$ reads "the set of real x such that $x \neq 2$". The colon (sometimes a vertical bar) means "such that".

Python programmers already know this: `range(2, 5)` is the half-open interval $[2, 5)$ of integers.

11.2.3. Why domains get restricted

When a function is given by a formula with no domain stated, the convention is to use the **natural domain**: every real number for which the formula makes sense. In real-number algebra, only a few things can go wrong:

1. **Division by zero.** A denominator must not be zero.
2. **Even roots of negatives.** $\sqrt{\ }$, $\sqrt[4]{\ }$, … need a non-negative argument.
3. **Logarithms of non-positives.** $\log(u)$ needs $u > 0$ (you will meet this in Chapter 12).

A domain may also be restricted **by the situation**: a length cannot be negative, a number of people must be a whole number, a time might run only from launch to landing. These are called **contextual** or **practical** domains.

Example 65. Procedure — finding the natural domain of a formula

Input: a formula $f(x)$.

Steps:

1. Start with all real numbers, $\mathbb{R}$.
2. For every denominator, solve "denominator $= 0$" and remove those values.
3. For every even root, require "inside ≥ 0" and solve the inequality.
4. For every logarithm, require "inside > 0" and solve.
5. Intersect all the conditions (an input must pass every test).
6. Write the result in interval notation.

Output: the natural domain of f.

11.2.4. Worked examples

Example 1. $g(x) = \dfrac{1}{x^2 - 4}$.

Denominator zero: $x^2 - 4 = 0 \iff (x-2)(x+2) = 0 \iff x = \pm 2$. No roots or logs. Domain: $(-\infty, -2) \cup (-2, 2) \cup (2, \infty)$.

Example 2. $f(x) = \dfrac{\sqrt{x-3}}{x-5}$.

$$\begin{aligned} \textbf{root condition: } & x - 3 \geq 0 \iff x \geq 3 \\ \textbf{denominator condition: } & x - 5 \neq 0 \iff x \neq 5 \\ \textbf{both: } & x \in [3, 5) \cup (5, \infty) \end{aligned}$$

Spot check: $f(3) = 0/(-2) = 0$ is fine; $f(5)$ would divide by zero; $f(2)$ would need $\sqrt{-1}$.

Example 3 (domain and range). $h(x) = \sqrt{4 - x^2}$.

Domain: $4 - x^2 \geq 0 \iff x^2 \leq 4 \iff -2 \leq x \leq 2$, so $[-2, 2]$.

Range: as x runs over $[-2, 2]$, x^2 runs over $[0, 4]$, so $4 - x^2$ runs over $[0, 4]$ and its square root runs over $[0, 2]$. The range is $[0, 2]$. (The graph is the upper half of the circle of radius 2, which you can check against Chapter 10.)

11.2.5. Finding the range

Ranges are harder than domains; there is no universal recipe. Three methods cover most school cases:

- **Reason about pieces**, as in Example 3: track what each operation does to a set of values.
- **Use a known graph.** A parabola $y = a(x - h)^2 + k$ with $a > 0$ has range $[k, \infty)$; with $a < 0$, $(-\infty, k]$ (see Chapter 8).
- **Solve for x.** Set $y = f(x)$ and ask: for which y does this equation have a solution x in the domain? Those y form the range.

Example of the third method: the range of $f(x) = \dfrac{x - 1}{x - 2}$.

$$
\begin{aligned}
y &= \frac{x - 1}{x - 2} \\
y(x - 2) &= x - 1 && \text{(multiply by } x - 2 \neq 0\text{)} \\
xy - 2y &= x - 1 \\
xy - x &= 2y - 1 && \text{(collect } x \text{ terms)} \\
x(y - 1) &= 2y - 1 \\
x &= \frac{2y - 1}{y - 1} && \text{(possible only if } y \neq 1\text{)}
\end{aligned}
$$

Every $y \neq 1$ gives a valid x (and that x is never 2, because $x = 2$ would need $2y - 2 = 2y - 1$, impossible). So the range is $\mathbb{R} \setminus \{1\}$. Hold on to this calculation: it is also how you find an inverse function, in Section 11.7.



Domain errors that hide inside simplification

$\dfrac{x^2 - 1}{x - 1}$ simplifies to $x + 1$, but the two are **not** the same function: the first is undefined at $x = 1$; the second is not. Always find the domain from the **original** formula, before simplifying. The graph of the first is the line $y = x + 1$ with a single missing point (a "hole") at $(1, 2)$. We return to holes in Section 11.9.

Computer-science connection: domains, types and partial functions

A function's domain is its **parameter type** plus any **preconditions**; its codomain is its **return type**. Mathematicians call a rule that is defined on only part of its stated input set a **partial function**. Most real code is partial: `math.sqrt` on `float`, list indexing on `int`, division on `int`. Languages handle partiality in different ways: raising an exception (Python), returning `NaN` (IEEE floats: `float('inf') - float('inf')`), returning an optional value (`Option<T>` in Rust, `Maybe` in Haskell), or making the domain a tighter type (a `NonZero` integer type). Each of these is a way of being honest about the domain.

US curriculum

Domain and range appear in **Algebra I**; the natural-domain procedure (especially with roots and rational expressions) and interval notation are central in **Algebra II**. "Codomain" is usually not taught until university; US school texts use "range" for what is strictly the image.

11.3. Graphs and the vertical line test

11.3.1. The problem

A table of values shows a few points. A formula hides behaviour inside algebra. The **graph** shows everything at once: where the function is positive, where it rises, where it turns, where it is undefined. How do you know whether a curve someone draws is the graph of a function at all?

11.3.2. Vocabulary

Graph of a function

The **graph** of f is the set of all points $(x, f(x))$ for x in the domain. In set-builder form: $\{(x, y) : x \in \text{domain},\ y = f(x)\}$.

So "the point (a, b) is on the graph" means exactly "$f(a) = b$". Reading a graph is evaluating a function with your eyes: go along to $x = a$, then up or down to the curve, and read off the height.

Intercepts, zeros and roots

- The **y-intercept** is where the graph crosses the y-axis: the point $(0, f(0))$. A function has at most one.
- An **x-intercept** is where the graph meets the x-axis: a point $(a, 0)$ with $f(a) = 0$.
- A number a with $f(a) = 0$ is a **zero** of the function, also called a **root** (of the equation $f(x) = 0$). Zeros of f are the x-coordinates of its x-intercepts.

11.3.3. The vertical line test

Since each input x has exactly one output, the graph contains exactly one point above (or below) each x in the domain, and none above x values outside it. A vertical line is the set of all points with a fixed x, so:

Vertical line test: a curve is the graph of a function of x $\iff$ no vertical line meets it more than once.

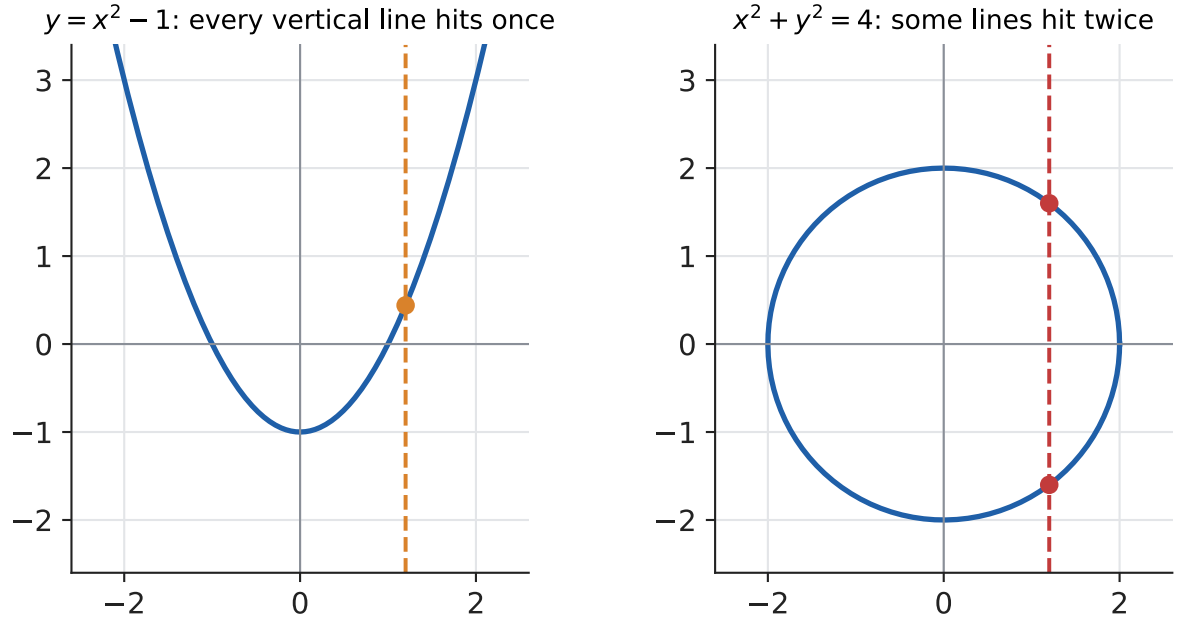


Figure 85. The parabola $y = x^2 - 1$ passes the vertical line test. The circle $x^2 + y^2 = 4$ fails: the line $x = 1.2$ meets it twice, at $y = \pm 1.6$.

The circle is still a perfectly good curve, described by an **equation** in x and y. It just is not the graph of one function of x. Solving for y gives $y = \pm\sqrt{4 - x^2}$: **two** functions, the upper semicircle $y = \sqrt{4 - x^2}$ (Example 3 above) and the lower semicircle $y = -\sqrt{4 - x^2}$. Splitting a curve into function-pieces like this is a routine move in calculus.

11.3.4. Reading a graph: what to look for

When you meet the graph of a new function, ask in this order:

1. **Domain and range.** What horizontal and vertical extent does it cover? Any gaps?
2. **Intercepts.** Where does it cross the axes?
3. **Sign.** Where is it above the x-axis (positive), where below (negative)?
4. **Direction.** Where does it rise, where fall? Where does it turn? (See Section 11.8.)
5. **Symmetry.** Is it mirror-symmetric about the y-axis, or symmetric under a half turn about the origin?
6. **Ends and breaks.** What happens as $x \to \pm\infty$? Are there asymptotes or holes (Section 11.9)?

Draw-and-test

Sketch any curve freehand on a coordinate grid. A vertical line follows the cursor and lights red whenever it meets the curve more than once; the app then offers to split the curve into the fewest function-pieces. Try a circle, a sideways parabola $x = y^2$, and the letter S.

US curriculum

Graphs of functions and the vertical line test are **Algebra I** topics (standard F-IF.5, F-IF.7). "Key features" (intercepts, intervals of increase, end behaviour) are emphasised in **Algebra II** and **Precalculus**.

11.4. Families of functions

11.4.1. The problem

There are infinitely many functions, but a small number of **families** do most of the work in science and computing. If you know the shape of each family's basic member (its **parent function**), you can sketch almost anything by shifting and stretching (the next section). This section is a field guide.

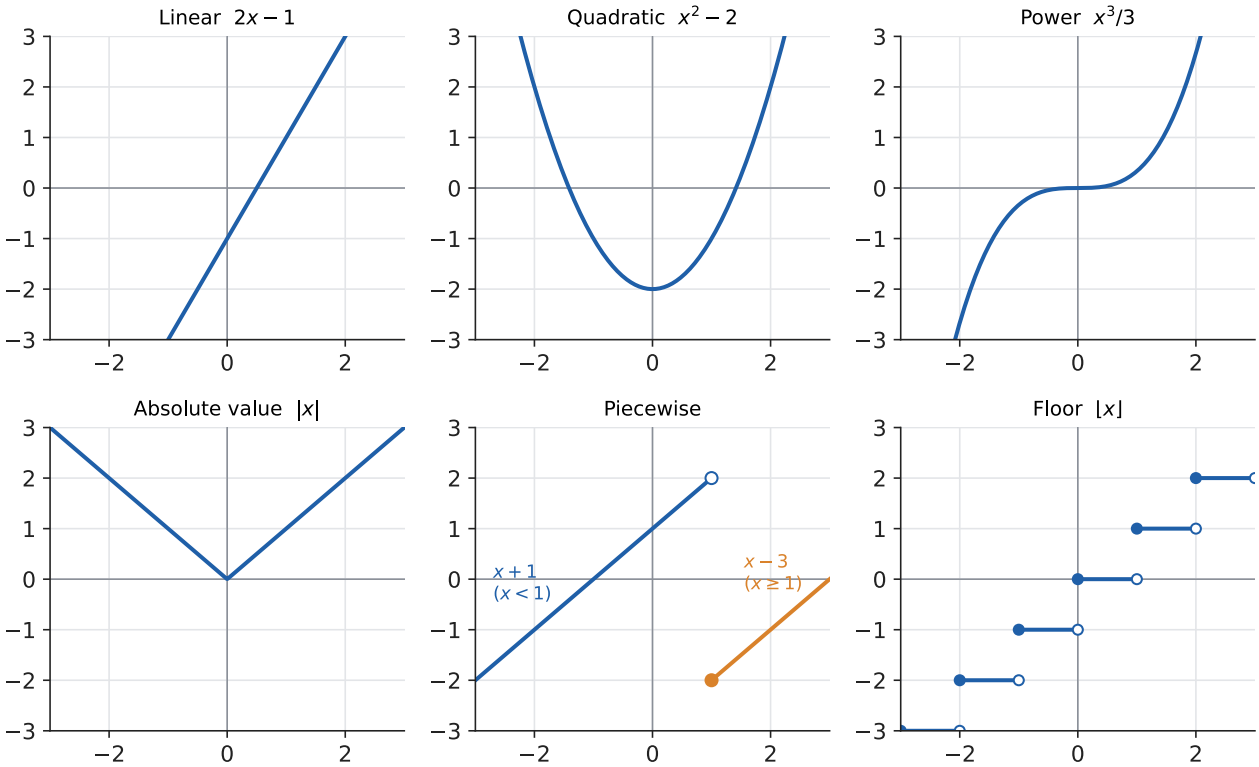


Figure 86. Parent functions from six families. Open circles mark points not included; filled circles mark points that are.

11.4.2. Linear functions

$f(x) = mx + b$. Graph: a straight line with slope m and y-intercept b (all developed in Chapter 5). The defining property: **equal steps in x give equal steps in y**. Increasing x by 1 always adds m. Linear functions model anything with a constant rate: a taxi's per-kilometre charge, a constant-speed journey, a linear cost.

11.4.3. Quadratic functions

$f(x) = ax^2 + bx + c$ with $a \neq 0$. Graph: a parabola, opening up if $a > 0$ and down if $a < 0$, with a vertex at $x = -b/(2a)$ (see Chapter 8). Here equal steps in x give **linearly growing** steps in y: for x^2 the outputs 0, 1, 4, 9, 16 have differences 1, 3, 5, 7. Projectiles, areas, and the running time of a double loop are quadratic.

11.4.4. Power functions

$f(x) = x^n$ (or kx^n). The shape depends on n:

- **Even positive n** (2, 4, 6, ...): U-shaped, symmetric about the y-axis, range $[0, \infty)$. Higher powers are flatter near 0 and steeper beyond $|x| = 1$.
- **Odd positive n** (1, 3, 5, ...): S-shaped through the origin, range $\mathbb{R}$.
- $n = -1$: the **reciprocal function $1/x$**, two separate branches, undefined at 0 (see Section 11.9).
- $n = 1/2$: the **square root function $\sqrt{x}$**, domain and range $[0, \infty)$, the right half of a sideways parabola.

Power laws appear everywhere: $A = \pi r^2$, $V = \frac{4}{3}\pi r^3$, gravity proportional to r^{-2}, algorithm costs n^2, n^3.

11.4.5. Absolute value



Absolute value function

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

It is the distance from x to 0 on the number line, so it is never negative. $|x - a|$ is the distance between x and a. The graph is a V with its corner at the origin.

The definition above is our first **piecewise** formula, and the "$-x$" branch is the one that surprises people: if $x = -3$ then $-x = 3$. The minus sign makes a negative number positive.

11.4.6. Piecewise functions



Piecewise function

A **piecewise function** uses different formulas on different parts of its domain:

$$f(x) = \begin{cases} x + 1 & \text{if } x < 1 \\ x - 3 & \text{if } x \geq 1 \end{cases}$$

The big brace is read "$f(x)$ equals $x + 1$ if $x < 1$, and $x - 3$ if $x \geq 1$". The conditions must not overlap (else some x would get two outputs) and together should cover the whole domain. On a graph, a **filled** dot shows an endpoint that belongs to a piece; an **open** dot shows one that does not.

The example above **jumps** at $x = 1$: approaching from the left, the values approach 2; at $x = 1$ itself, the value is $1 - 3 = -2$. A jump like this is a **discontinuity**, a concept made precise in Volume 2, Chapter 16 .

Piecewise functions are just `if` / `elif` / `else` written in mathematics. Every function in your code with a branch is piecewise.

11.4.7. Step functions: floor, ceiling and rounding



Floor and ceiling

- The **floor** $\lfloor x \rfloor$ is the greatest integer less than or equal to x: "round down".
- The **ceiling** $\lceil x \rceil$ is the least integer greater than or equal to x: "round up".

$\lfloor 3.7 \rfloor = 3$, $\lceil 3.2 \rceil = 4$, $\lfloor 5 \rfloor = \lceil 5 \rceil = 5$. Careful with negatives: $\lfloor -3.5 \rfloor = -4$ (down means towards $-\infty$, not towards 0), and $\lceil -3.5 \rceil = -3$.

These are **step functions**: their graphs are flat stairs with a jump at each integer.



Floor is not truncation

"Chop off the decimals" (**truncation**, rounding towards zero) agrees with floor for positive numbers and with ceiling for negative numbers. Languages disagree about which one integer division uses. Python's `//` is floor division; C, Java and JavaScript's `Math.trunc` truncate. Rounding has its own trap: Python 3's `round` uses **round half to even** ("banker's rounding"), so `round(2.5)` is 2 while `round(3.5)` is 4.

PYTHON

```
import math

def pages_needed(n_items, per_page):
    return math.ceil(n_items / per_page)

print(pages_needed(95, 10), pages_needed(100, 10), pages_needed(101, 10))
print(7 // 2, -7 // 2, int(-7 / 2))        # floor vs truncation
print(math.floor(-3.5), math.ceil(-3.5), round(2.5), round(3.5))
```

```
10 10 11
3 -4 -3
-4 -3 2 4
```

Computer-science connection: ceiling everywhere

Ceiling answers "how many containers do I need?": pages for n results at k per page ($\lceil n/k \rceil$), bytes for b bits ($\lceil b/8 \rceil$), blocks for a file, rounds of a tournament. Floor answers "which container is this in?": item i (counting from 0) is on page $\lfloor i/k \rfloor$. The integer-only idiom `(n + k - 1) // k` computes $\lceil n/k \rceil$ for non-negative integers without touching floating point, which matters when n is too large for a `float` to represent exactly.

11.4.8. Where you'd use this

- **Linear:** a plumber charges a $60 call-out fee plus $45 an hour: $C(t) = 60 + 45t$, so a 2.5-hour job is $60 + 112.5 = \$172.50$.
- **Quadratic:** braking distance grows with the **square** of speed. The UK Highway Code gives a braking distance of 14 m at 30 mph and 55 m at 60 mph: double the speed, roughly four times the distance ($55/14 \approx 3.9$). The thinking distance (9 m and 18 m) is linear in speed, so total stopping distance is a quadratic plus a linear term.
- **Power:** light, sound and gravity all follow an **inverse-square law**, $I = k/r^2$. Move a lamp from 1 m to 2 m away from your book and the light on the page falls to $1/2^2 = 1/4$ of what it was; to 3 m, $1/9$.
- **Absolute value:** engineering tolerances are absolute-value inequalities. A shaft specified as 25.00 ± 0.05 mm passes inspection when $|d - 25.00| \le 0.05$. In statistics and machine learning, the **mean absolute error** averages $|\text{prediction} - \text{truth}|$.

- **Piecewise:** a thermostat (heating on below 19 °C, off above 21 °C), income tax bands (Section 11.10), and the most common function in deep learning, the **ReLU** $\mathrm{ReLU}(x) = \max(0, x)$, which is 0 for $x < 0$ and x for $x \geq 0$. A neural network built from ReLUs is one enormous piecewise-linear function.
- **Step:** postage by weight band, car-park charges "per hour or part hour", phone-data bundles, and every pagination or bucket count in code (the CS sidebar above).

11.4.9. Choosing a family

When you have data or a description and need a model, the family is chosen by **how the output changes when the input changes**:

Pattern of change	Family	Typical example
Constant additive change per step	Linear	hourly wages
Additive change that itself grows steadily	Quadratic	distance fallen, double loops
Doubling or halving per step	Exponential (Chapter 12)	compound interest, cell growth
Scaling input by k scales output by k^n	Power	area and volume vs length
Different rules on different ranges	Piecewise	tax, shipping, thermostats
Flat, then sudden jumps	Step	postage, pagination
Repeating	Periodic (Chapter 13)	seasons, waves, clocks



Family portraits

Choose a family and drag sliders for its parameters (m, b for lines, a, b, c for quadratics, k, n for powers). A second panel shows the table of values and its first and second differences, so you can see "constant first differences = linear, constant second differences = quadratic" happening live.

Try it: Quadratic family with table

US curriculum

Linear and quadratic families: **Algebra I**. Absolute value and piecewise functions: **Algebra I** and **Algebra II**. Power, square root and step functions: **Algebra II**. Comparing families by their rate of change is Common Core standard F-LE (linear, quadratic and exponential models).

11.5. Transformations of graphs

11.5.1. The problem

You know the graph of $y = x^2$. What does $y = -2(x-3)^2 + 4$ look like? Or $y = |2x+6| - 1$? You could plot a table of points each time, but there is a far better way: every one of these is the parent graph **moved and stretched**, and the formula tells you exactly how.

11.5.2. Intuition: four moves

Start with a function f whose graph you know. There are four basic changes, two to the output and two to the input:

Formula	Change	Effect on graph
$f(x) + k$	add to output	shift **up** by k
$a \cdot f(x)$	multiply output	stretch **vertically** by factor a; if $a < 0$, also reflect in the x-axis
$f(x - h)$	subtract from input	shift **right** by h
$f(bx)$	multiply input	squeeze **horizontally** by factor b (stretch by $1/b$); if $b < 0$, also reflect in the y-axis

The output changes do what you expect. The input changes feel backwards: $x - 3$ moves the graph **right**, and $2x$ makes it **narrower**. Why?

11.5.3. Why it works: the input changes are "backwards"

Let $g(x) = f(x-3)$. Ask: where does g do what f does at 0? That is, when is the input to f equal to 0? When $x - 3 = 0$, so at $x = 3$. Everything f did at 0, g does at 3; everything f did at 1, g does at 4. The whole graph has moved 3 to the **right**. The subtraction is the delay: g has to wait until x reaches 3 before it gets the input 0.

Similarly, let $g(x) = f(2x)$. When is the input to f equal to 6? When $2x = 6$, so at $x = 3$. g gets through f's inputs **twice as fast**, so its graph is squashed horizontally by a factor of 2.

The rule of thumb: **changes outside f act on y directly; changes inside f act on x inversely.** To find where a point goes, solve "inside = old input" for x.

11.5.4. The general rule

All four moves combine into one formula:

$$g(x) = a\,f\big(b(x-h)\big) + k$$

If (x_0, y_0) is on the graph of f, then the corresponding point on the graph of g is where the input to f is again x_0:

$$\begin{aligned} b(x-h) &= x_0 && \text{(make the input to } f \text{ equal } x_0\text{)} \\ x &= \frac{x_0}{b} + h && \text{(divide by } b,\ \text{add } h\text{)} \\ y &= a\,f(x_0) + k = ay_0 + k \end{aligned}$$

$$\boxed{(x_0,\ y_0) \longmapsto \left(\frac{x_0}{b} + h,\ \ ay_0 + k\right)}$$

On the x side: scale by $1/b$, then shift by h. On the y side: scale by a, then shift by k. In each direction, **scale first, then shift**.

Example 66. Procedure — sketching $g(x) = a\,f(b(x-h)) + k$

Input: a parent function f with known graph; a formula for g.

Steps:

1. Rewrite the inside in the form $b(x-h)$ by factoring out the coefficient of x. ($f(2x-6)$ becomes $f(2(x-3))$, so $h = 3$, not 6.)
2. Read off a, b, h, k. Write $x+3$ as $x-(-3)$, so $h = -3$.
3. Choose 3–5 easy key points (x_0, y_0) on the parent (vertex, intercepts, corners).
4. Map each with $(x_0, y_0) \mapsto (x_0/b + h,\ ay_0 + k)$.
5. Plot the new points and draw the parent shape through them.
6. Check by evaluating g directly at one point.

Output: a sketch of g and its key points.

11.5.5. Worked example 1: a reshaped parabola

Sketch $g(x) = -2(x-3)^2 + 4$. *Try it:* Graph g(x) = −2(x − 3)² + 4 .

Parent: $f(x) = x^2$. Here $a = -2$, $b = 1$, $h = 3$, $k = 4$. Key points on the parent: $(-1, 1), (0, 0), (1, 1), (2, 4)$.

$$\begin{aligned} (-1, 1) &\mapsto (-1+3,\ -2\cdot 1 + 4) = (2, 2) \\ (0, 0) &\mapsto (0+3,\ -2\cdot 0 + 4) = (3, 4) && \text{(the vertex)} \\ (1, 1) &\mapsto (1+3,\ -2\cdot 1 + 4) = (4, 2) \\ (2, 4) &\mapsto (2+3,\ -2\cdot 4 + 4) = (5, -4) \end{aligned}$$

So: a parabola opening **down** (since $a < 0$), twice as steep as x^2, with vertex $(3, 4)$. Check directly: $g(5) = -2(2)^2 + 4 = -8 + 4 = -4$. Correct. This is exactly the **vertex form** of a quadratic from Chapter 8; now you can see that vertex form **is** the transformation rule.

11.5.6. Worked example 2: a horizontal squeeze in disguise

Sketch $g(x) = |2x + 6| - 1$. *Try it:* Graph g(x) = |2x + 6| − 1 .

Step 1, factor the inside: $2x + 6 = 2(x + 3) = 2(x - (-3))$. So $a = 1$, $b = 2$, $h = -3$, $k = -1$. Parent key points of $|x|$: $(-2, 2), (0, 0), (2, 2)$.

$$\begin{aligned} (-2, 2) &\mapsto \left(\tfrac{-2}{2} - 3,\ 2 - 1\right) = (-4, 1) \\ (0, 0) &\mapsto \left(\tfrac{0}{2} - 3,\ 0 - 1\right) = (-3, -1) \qquad \textbf{(the corner)} \\ (2, 2) &\mapsto \left(\tfrac{2}{2} - 3,\ 2 - 1\right) = (-2, 1) \end{aligned}$$

A V with its corner at $(-3, -1)$, with sides of slope ± 2. Check: $g(-2) = |2| - 1 = 1$. Correct.

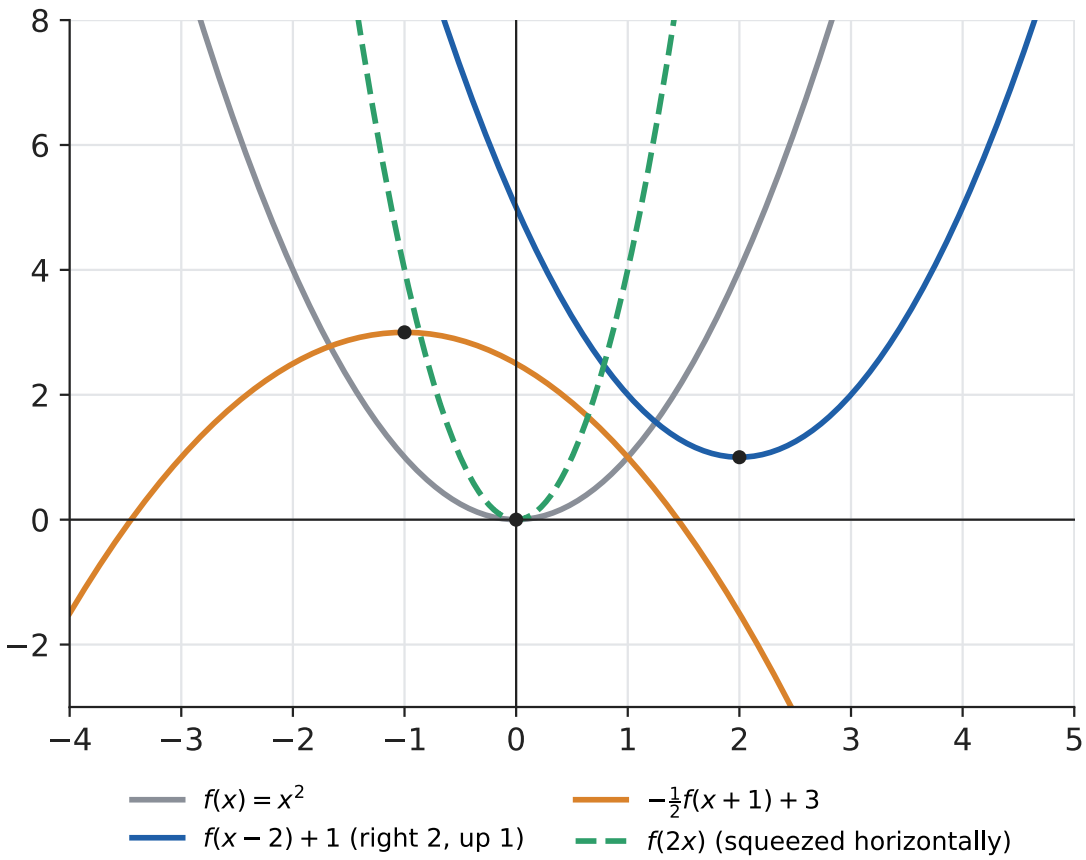


Figure 87. Transforming $f(x) = x^2$. The vertex $(0, 0)$ moves to $(2, 1)$ under $f(x-2)+1$ and to $(-1, 3)$ under $-\frac{1}{2}f(x+1)+3$. The dashed curve $f(2x)$ is the parent squeezed horizontally by a factor of 2.

```python
def transform(points, a=1, b=1, h=0, k=0):
    """Image of points on y=f(x) under y = a*f(b(x-h)) + k."""
    return [(x / b + h, a * y + k) for x, y in points]

base = [(-1, 1), (0, 0), (1, 1), (2, 4)]       # points on y = x^2
print(transform(base, a=-2, b=1, h=3, k=4))
print(transform(base, b=2))
```

```
[(2.0, 2), (3.0, 4), (4.0, 2), (5.0, -4)]
[(-0.5, 1), (0.0, 0), (0.5, 1), (1.0, 4)]
```



Shift by h, not by the constant

- $f(2x-6)$ is shifted right by **3**, not 6: factor first, $f(2(x-3))$. Doing the shift by 6 then the squeeze by 2 lands everything in the wrong place.
- $f(x+3)$ moves **left** 3. The sign inside is the opposite of the direction.
- $-f(x)$ reflects in the x-axis (flips up/down); $f(-x)$ reflects in the y-axis (flips left/right). They are different unless the function has symmetry.

11.5.7. Where you'd use this

The four moves have everyday meanings once the input is **time**:

- $f(t-h)$ is a **delay**. If $f(t)$ is the temperature in London at time t (hours), a city whose weather pattern arrives 3 hours later sees $f(t-3)$. An echo of a sound $f(t)$ arriving 0.25 s later at 30% of the volume is $0.3\,f(t-0.25)$; audio software builds reverb by adding many such copies.
- $f(bt)$ is a **speed change**. Playing a video at 2× speed shows $f(2t)$: everything happens at half the time. (Played back at 2× without correction, audio frequencies double too, which is why sped-up voices sound squeaky.)
- $af(t)+k$ is a **change of units or calibration**. Converting a temperature graph from Celsius to Fahrenheit is $1.8f(t)+32$: a vertical stretch by 1.8 and a shift up by 32. A bathroom scale that reads 0.5 kg heavy needs $f-0.5$.

In science, recognising a transformed parent saves work: once you know the shape of $y=x^2$, you know the shape of every projectile path, and once you know $1/x$ you know Boyle's law ($P=k/V$) for every gas sample.

Computer-science connection: transforms are coordinate changes

Every graphics system does the same thing. To draw a chart, you map "data coordinates" to "screen coordinates" by a scale and a shift in each direction: $x_{\text{px}} = s_x(x - x_{\min})$, $y_{\text{px}} = H - s_y(y - y_{\min})$ (the $H - \ldots$ flips the axis, because screen y points down, as discussed in Chapter 10). Animation easing curves are a parent curve rescaled to the animation's start, end and duration. In Volume 2, Chapter 27 , scaling, reflecting and rotating are packaged as matrices, and shifting is added with a trick called homogeneous coordinates.

Four sliders

Pick a parent (x^2, $|x|$, $\sqrt{x}$, x^3, $\lfloor x \rfloor$) and drag sliders for a, b, h, k in $af(b(x-h))+k$. Ghost arrows show where each key point of the parent goes. Then try "guess the formula": the app shows a transformed graph and you set the sliders to match it.

Try it: a f(b(x − h)) + k sliders

US curriculum

Transformations of graphs are Common Core standard F-BF.3, introduced in **Algebra I** for linear, quadratic and absolute value functions, generalised in **Algebra II**, and applied to trigonometric functions in **Precalculus**.

11.6. Composition: functions in a pipeline

11.6.1. The problem

A shop converts a price in dollars to euros, then adds 20% tax. Converting is one function; adding tax is another. The whole process (dollars in, final euro price out) is a third function made by running the first, then feeding its output into the second. That is **composition**, and it is how complicated functions are built from simple ones.

11.6.2. Vocabulary



Composition of functions

The **composition** of f and g is the function $f \circ g$, read "f composed with g", "f after g", or "f of g":

$$(f \circ g)(x) = f\big(g(x)\big)$$

Order: g is applied **first**, even though it is written second. Read $f \circ g$ from right to left, like nested function calls `f(g(x))` in code. The small circle $\circ$ is not a multiplication dot.

$f \circ g$ is defined at x only when x is in the domain of g **and** $g(x)$ is in the domain of f.

11.6.3. Worked example 1: order matters

Let $f(x) = x^2$ and $g(x) = x + 3$.

$$(f \circ g)(x) = f\big(g(x)\big) = f(x+3) = (x+3)^2 = x^2 + 6x + 9$$
$$(g \circ f)(x) = g\big(f(x)\big) = g(x^2) = x^2 + 3$$

At $x = 2$: $(f \circ g)(2) = 25$ but $(g \circ f)(2) = 7$. "Add 3 then square" is not "square then add 3". Composition is **not commutative**. (It is, however, **associative**: $f \circ (g \circ h) = (f \circ g) \circ h$, both meaning "do h, then g, then f".)

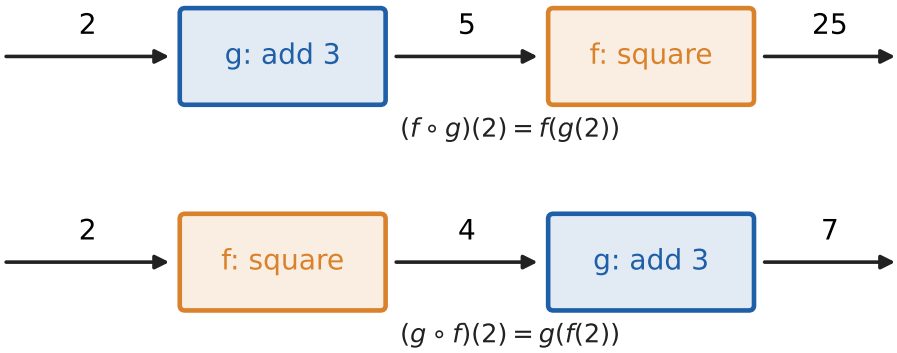


Figure 88. The same two machines in opposite orders. The value flows left to right, so $f \circ g$ runs g first.

11.6.4. Worked example 2: the domain of a composition

Let $f(x) = \sqrt{x}$ and $g(x) = x - 4$.

- $(f \circ g)(x) = \sqrt{x - 4}$. Need $g(x) = x - 4 \geq 0$, so the domain is $[4, \infty)$.
- $(g \circ f)(x) = \sqrt{x} - 4$. Need $x \geq 0$ for the root; then subtracting 4 is always fine. Domain $[0, \infty)$.

A subtler case: with $p(x) = x^2$ and $q(x) = \sqrt{x}$, $(p \circ q)(x) = (\sqrt{x})^2 = x$. The simplified formula x looks defined everywhere, but $p \circ q$ is defined only for $x \geq 0$, because q must be able to act first. As with holes, **find the domain before simplifying**.

11.6.5. Decomposition

Often the useful move is the reverse: seeing a complicated function as a composition of simple ones.

$h(x) = (3x - 1)^5$ is $f(g(x))$ with **inner** function $g(x) = 3x - 1$ and **outer** function $f(u) = u^5$.

$k(x) = \sqrt{x^2 + 1}$ is $f(g(x))$ with $g(x) = x^2 + 1$ and $f(u) = \sqrt{u}$.

The question to ask is "what would I compute **last** on a calculator?" That is the outer function. This skill is exactly what the **chain rule** needs in Volume 2, Chapter 17 , where the derivative of a composition is built from the derivatives of the pieces.

The transformation rule of the last section is itself a composition: $af(b(x - h)) + k$ is "subtract h, multiply by b, apply f, multiply by a, add k", a five-stage pipeline.

11.6.6. Where you'd use this

- **Shopping.** The opening problem with numbers: convert dollars to euros at 0.92, then add 20% tax. A \$50 item becomes $50 \times 0.92 = 46$ euros, then $46 \times 1.2 = 55.20$ euros. Here the order doesn't matter (both steps are multiplications), but with a 10%-off sale and a \$5 coupon it does: discount then coupon gives $0.9 \times 50 - 5 = \$40.00$; coupon then discount gives $0.9 \times (50 - 5) = \$40.50$. Shops choose the order, and now you know to check.

- **Science.** Unit conversions chain: km/h to m/s (divide by 3.6), then m/s to kinetic energy per kilogram ($v \mapsto \frac{1}{2}v^2$). A sensor reading passes through calibration, filtering and conversion before it reaches a screen; each stage is a function, and the whole instrument is their composition.
- **Machine learning.** A neural network is a long composition: layer 1, then layer 2, …, each a simple function. Training it uses the chain rule (Volume 2, Chapter 17), which is precisely "how does a composition change when its pieces change?"

Computer-science connection: composition as pipelines

Unix pipes (`cat log | grep ERROR | sort | uniq -c`) are function composition written left to right. So are method chains (`s.strip().lower().split()`) and data-processing pipelines (map, filter, reduce). Functional languages make composition an operator: Haskell's `.` is exactly $\circ$, and F# and Elixir have a left-to-right pipe `|>`. In Python you can write your own:

```python
from functools import reduce

def compose(*fs):
    """compose(f, g, h)(x) == f(g(h(x))) -- rightmost runs first."""
    return reduce(lambda f, g: lambda x: f(g(x)), fs, lambda x: x)

def f(x): return x ** 2
def g(x): return x + 3

print(compose(f, g)(2))   # f(g(2)) = (2+3)^2
print(compose(g, f)(2))   # g(f(2)) = 2^2 + 3

slugify = compose(lambda s: "-".join(s.split()), str.lower, str.strip)
print(slugify("  Functions and Their Graphs "))
```

```
25
7
functions-and-their-graphs
```

The starting value of the `reduce` is the **identity function** $x \mapsto x$, which does nothing; it plays the role for composition that 0 plays for addition and 1 for multiplication.

US curriculum

Composition is Common Core standard F-BF.1c, taught in **Algebra II**; decomposition is practised heavily in **Precalculus** and **AP Calculus** (chain rule).

11.7. Inverse functions

11.7.1. The problem

Fahrenheit from Celsius is $F = \frac{9}{5}C + 32$. A visitor wants the other direction: given F, what is C? We want a function that **undoes** the first one. When does such an undo function exist, and how do you find it?

11.7.2. Intuition: run the machine backwards

In the mapping picture, inverting a function means **reversing every arrow**. For the result to be a function again, two things must hold:

- No output may receive **two** arrows; otherwise, reversed, it would send out two arrows. (Squaring fails: 4 receives arrows from both 2 and −2, so "un-squaring 4" is ambiguous.)
- Every element of the codomain should receive an arrow; otherwise, reversed, it would send out none. (This is often fixed simply by shrinking the codomain to the range.)



One-to-one and inverse

A function is **one-to-one** (or **injective**) if different inputs always give different outputs: $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$. Equivalently, $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$.

If f is one-to-one with range R, its **inverse function** $f^{-1} : R \to$ **domain of** f is defined by

$$f^{-1}(y) = x \iff f(x) = y.$$

Read f^{-1} as "f inverse". The inverse undoes f and vice versa: $f^{-1}(f(x)) = x$ for every x in the domain of f, and $f(f^{-1}(y)) = y$ for every y in its range. In composition notation, $f^{-1} \circ f$ and $f \circ f^{-1}$ are identity functions. The domain and range swap roles.



f^{-1} is not $1/f$

The superscript -1 on a function name means **inverse function**, not reciprocal. If $f(x) = 2x$, then $f^{-1}(x) = x/2$ but $1/f(x) = 1/(2x)$. The notation is borrowed from composition: just as $a^{-1}a = 1$, $f^{-1} \circ f =$ **identity**. To make matters worse, trigonometry uses $\sin^2 x$ to mean $(\sin x)^2$ while $\sin^{-1} x$ means the inverse sine; Chapter 13 deals with that.

11.7.3. The horizontal line test

On a graph, "two inputs with the same output" means two points at the same height. So:

Horizontal line test: f **is one-to-one** $\iff$ **no horizontal line meets its graph more than once.**

Lines with nonzero slope pass. x^3 passes. x^2 fails (the line $y = 4$ meets it at $x = \pm 2$), as does $|x|$. Any function that is strictly increasing or strictly decreasing everywhere on its domain passes automatically (see Section 11.8).

11.7.4. Why the graph reflects in $y = x$

If (a, b) is on the graph of f, then $f(a) = b$, so $f^{-1}(b) = a$, so (b, a) is on the graph of f^{-1}. Swapping the coordinates of every point is the same as reflecting the plane in the line $y = x$. **The graph of f^{-1} is the mirror image of the graph of f in the line $y = x$.**

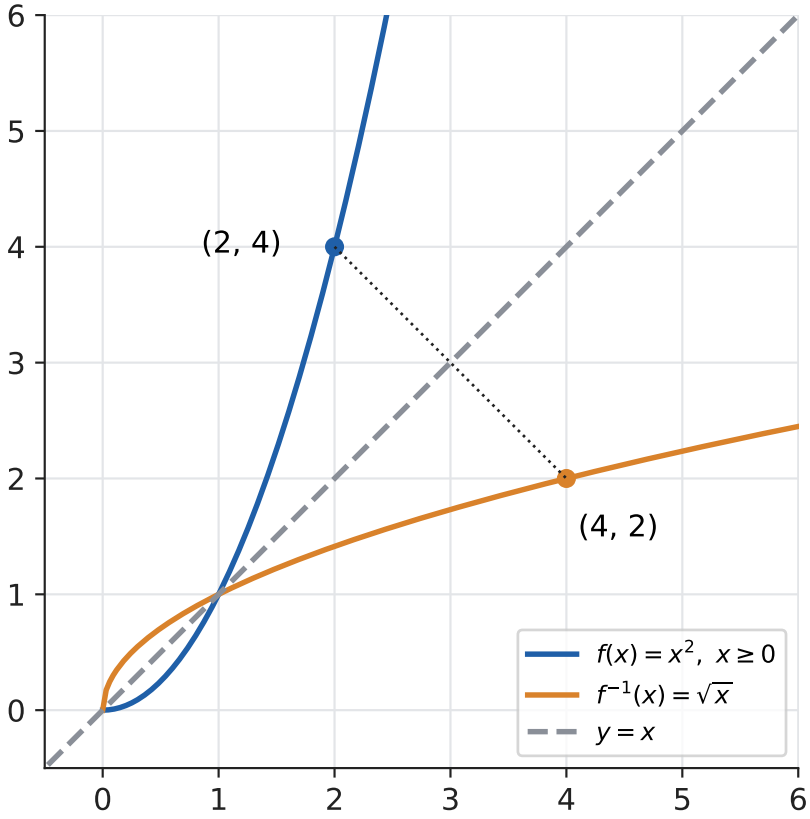


Figure 89. x^2 restricted to $x \geq 0$ and its inverse $\sqrt{x}$. The point $(2, 4)$ on one graph corresponds to $(4, 2)$ on the other; the dotted segment between them is bisected at right angles by $y = x$.

11.7.5. Restricting the domain

x^2 on all of $\mathbb{R}$ has no inverse. But on the restricted domain $[0, \infty)$ it is one-to-one, and its inverse is $\sqrt{x}$. On $(-\infty, 0]$ it is also one-to-one, with inverse $-\sqrt{x}$. The square root function exists **because** we chose the right half. The same choice gives the inverse trigonometric functions in Chapter 13, and the logarithm is the inverse of the exponential (which needs no restriction) in Chapter 12.

Example 67. Procedure — finding an inverse function algebraically

Input: a formula $y = f(x)$.

Steps:

1. Check that f is one-to-one (horizontal line test, or monotonicity). If not, restrict the domain.
2. Write $y = f(x)$.
3. Solve the equation for x in terms of y. (Note any values of y for which this is impossible: they are outside the range.)
4. Swap the letters x and y if you want the inverse written as a function of x. Name it $f^{-1}(x)$.
5. State the domain of f^{-1} (the range of f).
6. **Check** with a number: pick a, compute $b = f(a)$, and confirm $f^{-1}(b) = a$.

Output: f^{-1} with its domain.

Some books swap x and y first and then solve; the two orders are equivalent. Solving first keeps the meaning clearer ("given the output, recover the input"), which matters in applications where x and y stand for different quantities like Celsius and Fahrenheit.

11.7.6. Worked example 1: a linear function

Find the inverse of $f(x) = \dfrac{2x + 3}{5}$.

$$
\begin{aligned}
y &= \frac{2x + 3}{5} \\
5y &= 2x + 3 && \text{(multiply both sides by 5)} \\
5y - 3 &= 2x && \text{(subtract 3)} \\
x &= \frac{5y - 3}{2} && \text{(divide by 2)} \\
f^{-1}(x) &= \frac{5x - 3}{2} && \text{(rename } y \text{ as } x\text{)}
\end{aligned}
$$

Check: $f(6) = (12 + 3)/5 = 3$, and $f^{-1}(3) = (15 - 3)/2 = 6$. Correct. Notice the inverse undoes the steps **in reverse order**: f multiplies by 2, adds 3, divides by 5; f^{-1} multiplies by 5, subtracts 3, divides by 2. Like taking off socks and shoes: the last thing on is the first thing off. In symbols, $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.

For Celsius and Fahrenheit, the same steps turn $F = \frac{9}{5}C + 32$ into $C = \frac{5}{9}(F - 32)$.

11.7.7. Worked example 2: a rational function

Find the inverse of $f(x) = \dfrac{3x + 1}{x - 2}$, $x \neq 2$.

$$
\begin{aligned}
y &= \frac{3x+1}{x-2} \\
y(x-2) &= 3x+1 && \text{(multiply by } x-2\text{)} \\
xy-2y &= 3x+1 && \text{(expand)} \\
xy-3x &= 2y+1 && \text{(gather all } x \text{ terms on the left)} \\
x(y-3) &= 2y+1 && \text{(factor out } x\text{)} \\
x &= \frac{2y+1}{y-3} && (y \neq 3) \\
f^{-1}(x) &= \frac{2x+1}{x-3}, \quad x \neq 3
\end{aligned}
$$

The range of f is everything except 3, which becomes the domain of f^{-1}. Check: $f(3) = 10/1 = 10$; $f^{-1}(10) = 21/7 = 3$. Correct.

11.7.8. Alternative approaches and choosing a method

- **Reverse the steps** (as in the socks-and-shoes remark). Fastest when x appears **once** in the formula: list the operations applied to x, then undo them in reverse order.
- **Solve for x** (the procedure). Needed when x appears more than once, as in Example 2: gather the x terms and factor.
- **Graphically**, reflect in $y = x$. Useful for sketching, not for formulas.
- **Numerically**, when no formula exists. If f is increasing, you can find $f^{-1}(y)$ by bisection: search for the x with $f(x) = y$ the way binary search finds a key (Volume 2, Chapter 32). This is how many libraries compute inverses of complicated functions.

11.7.9. Where you'd use this

- **Working backwards from a target.** At 5% interest, money grows as $A = 1.05P$ in a year. To have $1,000 next year you need the inverse: $P = A/1.05 = 1000/1.05 \approx \$952.38$ today. "How much do I need to start with?" is always an inverse-function question.
- **Accident investigation.** A car braking with friction coefficient μ skids a distance $d = v^2/(2\mu g)$ from speed v (a mechanics result). Police reconstructing a crash measure the skid and invert: $v = \sqrt{2\mu g d}$. With $\mu = 0.7$, $g = 9.81$ m/s² and a 30 m skid, $v = \sqrt{2(0.7)(9.81)(30)} \approx 20.3$ m/s, about 73 km/h. The formula needs $v \geq 0$: the restriction to the right half of the parabola, exactly as with $\sqrt{x}$.
- **Measurement in general.** Instruments measure one thing and report another: a thermometer measures expansion, a GPS measures signal delay, a camera measures light. Turning the raw reading back into the quantity you want is applying an inverse function, and it only works if the original was one-to-one over the range in use.

Computer-science connection: inverses in software

Inverse pairs are everywhere: encode/decode, serialize/deserialize, compress/decompress, encrypt/decrypt, `str()` / `int()`. The mathematical requirement transfers exactly. Lossless compression must be one-to-one (two different files must not compress to the same bytes), and that requirement alone proves no lossless compressor can shrink **every** file (there are fewer short files than long ones). A cryptographic hash, by contrast, is deliberately **not** invertible in practice. A good round-trip test (`decode(encode(x)) == x`) is precisely the identity $f^{-1} \circ f = \mathrm{id}$ checked on samples, and property-based testing tools generate those samples for you.



Mirror in $y = x$

Draw a function's graph; the app draws its reflection in $y = x$ and colours it red wherever the reflection fails the vertical line test. Drag a "domain window" to restrict the original until the reflection becomes a valid function.

Try it: Reflect $y = x^2$ in $y = x$

US curriculum

Inverse functions are Common Core standard F-BF.4, introduced in **Algebra II** (solving $f(x) = c$ and finding simple inverses) and extended in **Precalculus** (verifying by composition, restricting domains, inverse trig functions). The term "one-to-one" is standard at school; "injective" usually waits for university.

11.8. Symmetry and direction: even, odd, increasing, decreasing

11.8.1. The problem

Some graphs have a symmetry you can exploit: compute half, get the other half free. Others always go up, which guarantees an inverse exists and allows binary search. These properties can be tested algebraically, without drawing.

11.8.2. Even and odd functions



Even and odd functions

- f is **even** if $f(-x) = f(x)$ for every x in the domain. Its graph is mirror-symmetric in the y-axis. Examples: x^2, x^4, $|x|$, $\cos x$.
- f is **odd** if $f(-x) = -f(x)$ for every x in the domain. Its graph is unchanged by a half turn (180° rotation) about the origin. Examples: x, x^3, $1/x$, $\sin x$.

Most functions are **neither**. The names come from powers: x^n is even exactly when n is even, and odd when n is odd.

Worked example. Classify $f(x) = x^4 - 3x^2$, $g(x) = x^3 + x$, $h(x) = x^2 + x$.

$$\begin{aligned} f(-x) &= (-x)^4 - 3(-x)^2 = x^4 - 3x^2 = f(x) && \Rightarrow \text{ even} \\ g(-x) &= (-x)^3 + (-x) = -x^3 - x = -(x^3 + x) = -g(x) && \Rightarrow \text{ odd} \\ h(-x) &= (-x)^2 + (-x) = x^2 - x \end{aligned}$$

For h, the result $x^2 - x$ is neither $h(x) = x^2 + x$ nor $-h(x) = -x^2 - x$ (try $x = 1$: $h(-1) = 0$ while $h(1) = 2$). So h is neither even nor odd. One counterexample is enough to rule a property out; to establish it, you need the algebra for **all** x.

```python
def parity(f, samples=(0.5, 1, 2, 3.7)):
    even = all(abs(f(-x) - f(x)) < 1e-12 for x in samples)
    odd = all(abs(f(-x) + f(x)) < 1e-12 for x in samples)
    return "even" if even else "odd" if odd else "neither"

print(parity(lambda x: x**4 - 3*x**2))
print(parity(lambda x: x**3 + x))
print(parity(lambda x: x**2 + x))
```

```
even
odd
neither
```



Sampling is evidence, not proof

The `parity` helper only **tests** finitely many points. A function could pass at those samples and fail elsewhere. Code like this is good for catching mistakes, not for establishing a property; for that you need the algebra above. This distinction (testing versus proving) is the subject of Volume 3, Chapter 22 .



Every function is even plus odd

Any function defined on a symmetric domain splits uniquely into an even part and an odd part:

$$f(x) = \underbrace{\frac{f(x) + f(-x)}{2}}_{\text{even}} + \underbrace{\frac{f(x) - f(-x)}{2}}_{\text{odd}}$$

For a polynomial, this just separates even and odd powers. For e^x it produces the **hyperbolic cosine and sine**, $\cosh x$ and $\sinh x$, which describe the shape of a hanging chain. The same splitting idea, generalised, is the start of Fourier analysis.

11.8.3. Increasing and decreasing

Increasing, decreasing, monotonic
On an interval I:

- f is **increasing** if $x_1 < x_2 \Rightarrow f(x_1) < f(x_2)$ for all $x_1, x_2 \in I$ (the graph rises left to right);
- f is **decreasing** if $x_1 < x_2 \Rightarrow f(x_1) > f(x_2)$;
- f is **monotonic** on I if it is one or the other throughout.

Some books say "strictly increasing" for this and "increasing" or "non-decreasing" for the version with $\leq$. A **local maximum** is a point higher than all nearby points; a **local minimum**, lower. They occur where a function switches between increasing and decreasing.

Example: $f(x) = x^2$ is decreasing on $(-\infty, 0]$ and increasing on $[0, \infty)$, with a minimum at 0. The parabola $-2(x-3)^2 + 4$ is increasing on $(-\infty, 3]$, decreasing on $[3, \infty)$, with a maximum of 4 at $x = 3$.

Two important facts:

1. **A strictly monotonic function is one-to-one**, and therefore has an inverse. (If $x_1 \neq x_2$, one is smaller, so their outputs differ.) This is why the restriction of x^2 to $[0, \infty)$ works.
2. Finding where a function increases and decreases, for functions you cannot just picture, is one of the main jobs of the derivative (Volume 2, Chapter 18).

Computer-science connection: monotonicity powers binary search

Binary search works on a sorted array because "the value at index i" is a non-decreasing function of i. More generally, you can binary-search the **answer** to any problem where a yes/no predicate is monotone: "can we finish all jobs with capacity c?" is false, false, ..., false, true, true, ... as c increases, so the smallest working capacity can be found in $O(\log n)$ checks. `git bisect` uses the same idea: "is the bug present at commit i?" is (assumed) monotone along history.

US curriculum

Even and odd functions and intervals of increase and decrease are **Algebra II** and **Precalculus** topics (F-BF.3 asks students to recognise even and odd functions from graphs and formulas). Monotonicity via derivatives is **AP Calculus**.

11.9. Rational functions and asymptotes: a first look

11.9.1. The problem

The average cost per item of a print run, with a fixed setup cost of \$500 and \$2 per copy, is $C(n) = \dfrac{500 + 2n}{n}$. For small runs the cost is enormous; for huge runs it settles towards \$2 but never gets there. Graphs like this, which blow up near some inputs and level off far away, belong to **rational functions**.



Rational function and asymptote

A **rational function** is a quotient of two polynomials, $R(x) = \dfrac{P(x)}{Q(x)}$, with Q not the zero polynomial (the name comes from **ratio**, as with rational numbers). Its natural domain is every real number except the zeros of Q.

An **asymptote** is a line that the graph approaches ever more closely:

- a **vertical asymptote** $x = a$: as x gets near a, $R(x)$ grows without bound (towards $+\infty$ or $-\infty$);
- a **horizontal asymptote** $y = L$: as $x \to \infty$ or $x \to -\infty$, $R(x)$ gets as close to L as you like.

"$x \to \infty$" reads "as x grows without bound". The precise language for "gets as close as you like" is the **limit**, defined in Volume 2, Chapter 16 .

11.9.2. Intuition: the reciprocal function

The parent is $y = 1/x$. Near $x = 0$, dividing 1 by a tiny number gives a huge number: $1/0.001 = 1000$, $1/(-0.001) = -1000$. So the graph shoots up on the right of 0 and down on the left: a vertical asymptote at $x = 0$. Far out, dividing 1 by a huge number gives almost 0: a horizontal asymptote at $y = 0$.

Now use transformations. $y = \dfrac{1}{x - 2} + 1$ is $1/x$ shifted right 2 and up 1, so its asymptotes are $x = 2$ and $y = 1$. And that is the function from the range example, since

$$\frac{x - 1}{x - 2} = \frac{(x - 2) + 1}{x - 2} = 1 + \frac{1}{x - 2}.$$

This explains, geometrically, why its range was everything except 1.

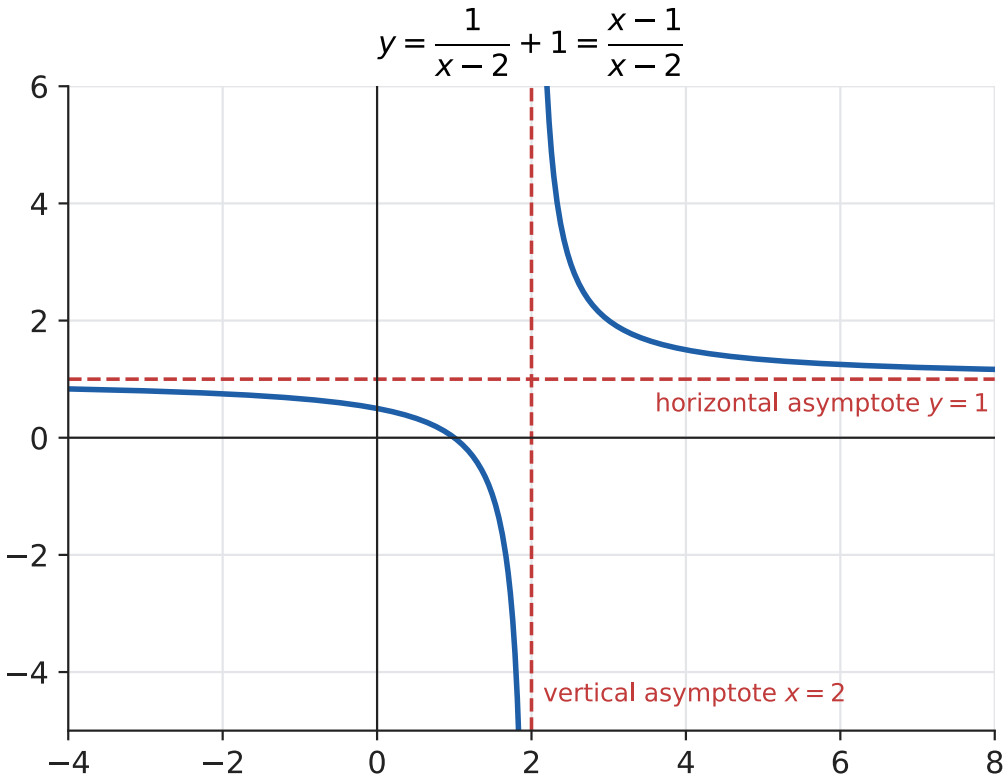


Figure 90. $y = (x-1)/(x-2)$ is the reciprocal graph moved right 2 and up 1. The asymptotes move with it.

11.9.3. The method

Example 68. Procedure — asymptotes and holes of a rational function $P(x)/Q(x)$

Input: a rational function $R(x) = P(x)/Q(x)$.

Steps:

1. Factor P and Q completely (see Chapter 7).
2. **Holes:** for each factor $(x-a)$ common to P and Q, cancel it; the graph has a hole at $x = a$. Its y-coordinate is the cancelled formula evaluated at a.
3. **Vertical asymptotes:** after cancelling, each remaining zero a of the denominator gives a vertical asymptote $x = a$.
4. **Horizontal asymptote:** compare degrees $\deg P$ and $\deg Q$:
   a. $\deg P < \deg Q$: $y = 0$;
   b. $\deg P = \deg Q$: $y = \dfrac{\text{leading coefficient of } P}{\text{leading coefficient of } Q}$;
   c. $\deg P > \deg Q$: no horizontal asymptote (if $\deg P = \deg Q + 1$ there is a slanted, or **oblique**, asymptote, found by polynomial division).

Output: holes, vertical asymptotes, and the horizontal (or oblique) asymptote.

Why the degree rule works. Far from the origin, the highest power dominates each polynomial. For $R(x) = \dfrac{2x^2+1}{x^2-9}$, divide top and bottom by x^2:

$$R(x) = \frac{2 + 1/x^2}{1 - 9/x^2} \longrightarrow \frac{2+0}{1-0} = 2 \quad \text{as } x \to \pm\infty.$$

11.9.4. Worked example

Analyse $R(x) = \dfrac{2x^2+1}{x^2-9}$.

1. Factor: $2x^2+1$ does not factor over the reals (it is always positive); $x^2 - 9 = (x-3)(x+3)$.
2. No common factors, so no holes.
3. Vertical asymptotes: $x = 3$ and $x = -3$.
4. Degrees equal (2 and 2): horizontal asymptote $y = 2/1 = 2$.

Also: $R(0) = 1/(-9) = -1/9$ (the y-intercept), and since the numerator is never zero there are no x-intercepts. R is even, since only even powers appear.

A hole. $S(x) = \dfrac{x^2-1}{x-1} = \dfrac{(x-1)(x+1)}{x-1}$. The factor $x-1$ cancels, so there is a hole at $x = 1$, at height $1+1 = 2$, and **no** vertical asymptote. The graph is the line $y = x+1$ with the point $(1, 2)$ removed.

For the print-run example, $C(n) = (500+2n)/n$ has equal degrees, leading coefficients 2 and 1, so the horizontal asymptote is $y = 2$: the cost per copy approaches \$2. Writing it as $C(n) = 2 + 500/n$ makes that obvious: the setup cost is spread ever more thinly.



Asymptotes are not walls

- A graph **can** cross a horizontal asymptote; the asymptote describes only the far-away behaviour. $\dfrac{x}{x^2+1}$ has asymptote $y = 0$ and crosses it at $x = 0$. A graph **cannot** cross a vertical asymptote, since the function is undefined there.
- A zero of the denominator does not always give an asymptote: if it cancels with the numerator, you get a hole instead. Always factor first.
- Graphing calculators and plotting libraries often draw a near-vertical line joining the two branches across an asymptote. That line is an artefact of "connect the dots", not part of the graph.

11.9.5. Where you'd use this

- **Travel time.** For a fixed 50 km trip, time is $t(v) = 50/v$, the reciprocal shape. At 110 km/h it takes $50/110 \times 60 \approx 27.3$ minutes; at 130 km/h, about 23.1 minutes. Driving 20 km/h faster saves only about 4 minutes, because the curve is flattening out. (Going from 30 to 50 km/h on the same trip

would save 40 minutes.)

- **Physics and chemistry.** Boyle's law $P = k/V$ (halve the volume of a gas, double its pressure), two resistors in parallel $R = \dfrac{R_1 R_2}{R_1 + R_2}$, and the thin-lens equation are all rational functions; their vertical asymptotes are physically meaningful (a lens focusing at infinity, a short circuit).
- **Business.** Any "fixed cost plus per-unit cost, divided by units" is the print-run curve. A $1,200 coffee machine that saves $3 a day pays for itself in 400 days; the average cost per cup keeps falling but never below the cost of the beans.

Computer-science connection: amortisation is a rational function

The print-run curve $2 + 500/n$ is the shape of every **amortised cost**: a fixed overhead (connection setup, JIT compilation, a cache warm-up, building an index) shared over n uses. It explains why batching requests helps enormously at first and then hardly at all; the curve flattens towards its asymptote. Performance graphs of throughput against load often show the other asymptote: as utilisation ρ of a queue approaches 1, average waiting time behaves like $\rho/(1-\rho)$, a rational function with a vertical asymptote at $\rho = 1$. That is why a server at 95% load feels much worse than one at 80%.

US curriculum

Rational functions and asymptotes are introduced in **Algebra II** (graphing simple cases) and developed fully in **Precalculus**. The limit-based definition of an asymptote belongs to **Calculus**.

11.10. Functions as models

11.10.1. The problem

A function is also a **claim about the world**: "tax depends on income like this", "height depends on time like that". Building such a model means choosing variables, choosing a family, fitting parameters, and (the step most often skipped) stating the domain where the model is valid.

11.10.2. Worked example 1: a piecewise tax schedule

A (simplified, fictional) tax system: 10% on income up to $10,000, and 20% on every dollar above $10,000. Write tax T as a function of income x and compute $T(25{,}000)$.

On the first $10,000 the tax is $0.1x$. Above that, the first $10,000 contributes $0.1 \times 10{,}000 = 1{,}000$, and the excess $x - 10{,}000$ is taxed at 20%:

$$T(x) = \begin{cases} 0.1x & 0 \le x \le 10{,}000 \\ 1{,}000 + 0.2(x - 10{,}000) & x > 10{,}000 \end{cases}$$

$$\begin{aligned} T(25{,}000) &= 1{,}000 + 0.2(25{,}000 - 10{,}000) \qquad \text{(second branch, since } 25{,}000 > 10{,}000\text{)} \\ &= 1{,}000 + 0.2 \times 15{,}000 \\ &= 1{,}000 + 3{,}000 = 4{,}000 \end{aligned}$$

Check continuity at the boundary: the first branch gives $T(10{,}000) = 1{,}000$, and the second branch, as x approaches 10,000 from above, approaches $1{,}000 + 0 = 1{,}000$. No jump. That is the meaning of **marginal** rates: moving into a higher bracket only raises the tax on the extra dollars, so earning more never leaves you with less. (A system with a **jump** in tax at the boundary would create exactly that perverse effect, which is why real systems are designed as continuous piecewise-linear functions.)

11.10.3. Worked example 2: a step-function price

A courier charges \$3.00 for the first kilogram (or part of one) and \$1.50 for each additional kilogram or part. Model the cost and price a 2.3 kg parcel.

"Or part" means round up. A parcel of weight w counts as $\lceil w \rceil$ kilograms, of which the first is charged at \$3.00:

$$C(w) = 3 + 1.5\big(\lceil w \rceil - 1\big), \qquad w > 0.$$

$$C(2.3) = 3 + 1.5(\lceil 2.3 \rceil - 1) = 3 + 1.5(3 - 1) = 3 + 3 = 6.00$$

Sanity checks: $C(0.4) = 3 + 1.5(1 - 1) = 3.00$; $C(1) = 3.00$; $C(1.01) = 3 + 1.5 = 4.50$.

11.10.4. Worked example 3: a projectile and its practical domain

A ball is thrown upwards from 1.5 m with speed 20 m/s. Ignoring air resistance, its height after t seconds is $h(t) = -4.9t^2 + 20t + 1.5$ metres. What is the practical domain, the maximum height, and the range?

The model only makes sense from the throw ($t = 0$) until the ball lands ($h(t) = 0$, $t > 0$). Using the quadratic formula (Chapter 8) on $4.9t^2 - 20t - 1.5 = 0$:

$$t = \frac{20 + \sqrt{20^2 + 4(4.9)(1.5)}}{2(4.9)} = \frac{20 + \sqrt{429.4}}{9.8} \approx \frac{20 + 20.722}{9.8} \approx 4.155.$$

(The other root is negative, before the throw, so we discard it.) The domain is $[0, 4.155]$ approximately.

The vertex is at $t = -b/(2a) = -20/(2 \times -4.9) \approx 2.041$ s, and

$$h(2.041) \approx -4.9(2.041)^2 + 20(2.041) + 1.5 \approx -20.41 + 40.82 + 1.5 \approx 21.9 \text{ m}.$$

So the range is about $[0, 21.9]$ metres. Outside the domain the formula still produces numbers ($h(6) \approx -55$), but they mean nothing: the ball does not burrow 55 m underground.

Extrapolation

A model fitted on data from one range of inputs can be badly wrong outside it. A straight line fitted to a child's height from ages 2 to 10 predicts a 40-year-old over 3 m tall. Always state the domain of validity, and be suspicious of any prediction that leaves it. Machine-learning models have the same failure under the name **distribution shift**.

Fit a family

Load a small data set (stopping distance against speed, average cost against batch size, height against time). Choose a family, drag its parameters to fit the points, and watch a residual plot. Then drag a slider extending the x-range beyond the data and see which fitted families stay sensible and which go wild.

Try it: Fit braking distances

US curriculum

Modelling with functions runs through all of high-school math under the Common Core "Modeling" strand; piecewise and step-function models appear in **Algebra I** and **Algebra II**, projectile models in **Algebra I** (quadratics).

11.11. Functions in computing

Programmers use the word "function" every day, and it means something close to, but not the same as, the mathematical idea. Knowing exactly where they differ makes you a better reader of mathematics and a better writer of code.

11.11.1. Mathematical functions versus programming functions

A mathematical function is **only** an input-to-output assignment. It does not "run", take time, print anything, or remember previous calls. A programming function (a subroutine, procedure or method) is a piece of code that may do all of those things.



Pure function and side effect

A programming function is **pure** if

1. its output depends **only** on its arguments (no reading of global state, clocks, random numbers, files or networks), and
2. it has no **side effects**: it changes nothing outside itself (no writing to globals, files, screens or its arguments).

A pure function is a mathematical function. Calling it twice with the same arguments must give the same result, so the call can always be replaced by its value (**referential transparency**).

PYTHON

```
total = 0

def add_impure(x):
    global total          # reads and writes state outside itself
    total += x
    return total

def add_pure(acc, x):
    return acc + x        # output depends only on inputs

print(add_impure(5), add_impure(5))   # same input, different outputs
print(add_pure(0, 5), add_pure(0, 5)) # same input, same output
```

```
5 10
5 5
```

`add_impure` fails the defining property of a function: the input 5 produced two different outputs. Mathematically, it is a function of **two** inputs (the argument and the hidden state), and making that hidden input explicit, as `add_pure` does, is the standard repair. Functional programming is largely the discipline of making every input explicit.

Other differences worth knowing:

- **Partiality.** Many program functions do not return for some inputs: they raise, loop forever, or crash. Mathematically these are partial functions (see Section 11.2).
- **Time and memory.** Mathematics ignores how long a function takes; computer science studies exactly that ($O(n^2)$ and friends).
- **Equality.** Two mathematical functions are equal if they have the same domain and give the same output for every input, however differently they are computed. Bubble sort and merge sort compute the **same function**.
- **Floating point.** `f(x)` in code computes an approximation; $0.1 + 0.2$ is not exactly 0.3 (see Volume 2, Chapter 32).

11.11.2. Functions as data: first-class functions

In mathematics, functions are objects you can pass around: composition takes two functions and returns a function; the derivative (Volume 2, Chapter 17) takes a function and returns a function. Languages with **first-class functions** allow the same: functions can be stored in variables, passed as arguments, and returned as results. A function that takes or returns functions is a **higher-order function**. `compose` , `map` , `filter` , `sorted(key=…)` , and decorators are all higher-order. The Python `lambda x: x ** 2` is a direct transcription of $x \mapsto x^2$.

11.11.3. Lookup tables: a function is its table

Mathematically, a function on a finite domain is nothing more than its table of input–output pairs. Computing can trade time for space by storing that table:

```python
from functools import lru_cache
calls = 0

@lru_cache(maxsize=None)
def slow_square(n):
    global calls
    calls += 1
    return n * n

for n in [3, 4, 3, 3, 4, 5]:
    slow_square(n)
print("calls:", calls)

# A finite function *is* a table.
SQUARES = {n: n * n for n in range(10)}
print(SQUARES[7])
```

```
calls: 3
49
```

Memoisation (here via `lru_cache`) builds the table lazily: six calls, but only three distinct inputs, so only three real computations. This is only safe for **pure** functions; memoising `add_impure` would silently change its behaviour. Precomputed tables appear in CRC checksums (a 256-entry table per byte), gamma correction for images, sine tables in old games and embedded systems, and dynamic programming generally (Volume 3, Chapter 24).

11.11.4. Hash functions

A **hash function** maps data of any size (strings, files, objects) to a fixed-size value, typically an integer in a fixed range. A hash table uses it to choose a bucket: $\mathbf{bucket}(k) = \mathbf{hash}(k) \bmod n$.

```python
import zlib

def bucket(key, n_buckets=8):
    return zlib.crc32(key.encode()) % n_buckets

words = ["apple", "banana", "cherry", "date", "elder", "fig",
         "grape", "honeydew", "kiwi", "lemon"]
table = {}
for w in words:
    table.setdefault(bucket(w), []).append(w)
for b in sorted(table):
    print(b, table[b])
```

```
0 ['apple', 'cherry']
1 ['grape']
2 ['date']
3 ['honeydew']
4 ['kiwi', 'lemon']
5 ['elder', 'fig']
7 ['banana']
```

(We used `zlib.crc32` because it is deterministic; Python's built-in `hash` for strings is deliberately randomised per process, so its output would differ every run.)

A hash must be a true **function**: the same key must always land in the same bucket, or lookups fail. It is never **one-to-one** when the domain is bigger than the codomain: ten words into eight buckets must produce at least one **collision** (two keys, one bucket). This is the **pigeonhole principle** (Volume 3, Chapter 24). Hash-table design is the art of making collisions rare and cheap, not impossible. Cryptographic hashes (SHA-256) go further: collisions certainly exist, but finding one should be computationally infeasible.

11.11.5. Injective, surjective, bijective

The one-to-one idea from Section 11.7 has two partners, and the three together classify functions precisely.



Injective, surjective, bijective

For $f : A \to B$:

- f is **injective** (**one-to-one**) if no two inputs share an output: $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$. "No collisions."
- f is **surjective** (**onto**) if every element of the codomain is hit: for every $y \in B$ there is an $x \in A$ with $f(x) = y$. Equivalently, range = codomain. "Nothing missed."
- f is **bijective** (a **one-to-one correspondence**) if it is both. "Perfect pairing."

A function has a two-sided inverse $f^{-1} : B \to A$ **exactly when it is bijective**.

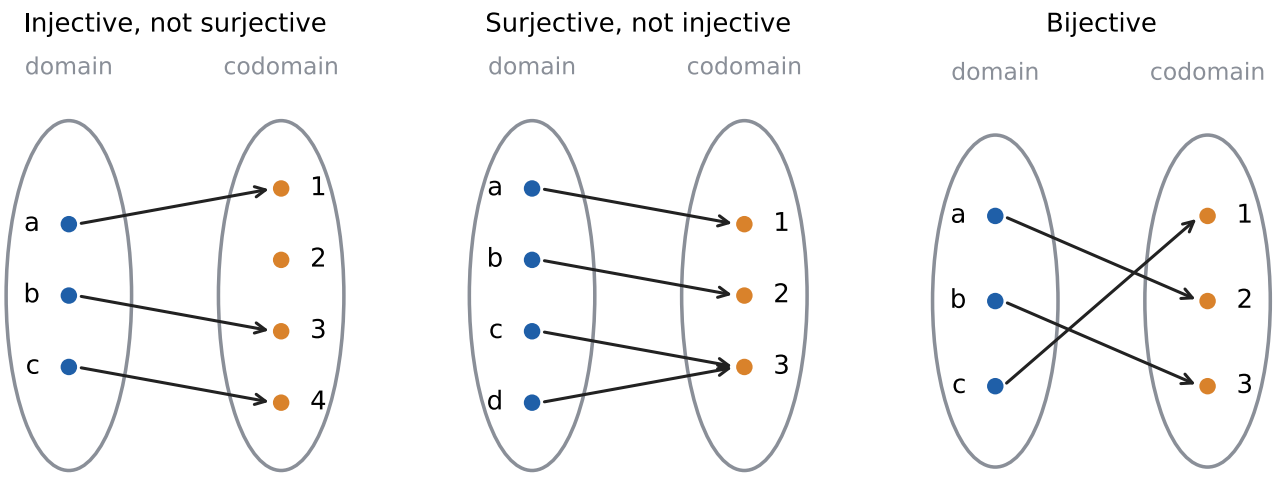


Figure 91. Arrow diagrams for the three properties. Injective: at most one arrow into each output. Surjective: at least one arrow into each output. Bijective: exactly one.

Surjectivity depends on the codomain you declare. $x \mapsto x^2$ is not surjective onto $\mathbb{R}$ (-1 is never hit) but is surjective onto $[0, \infty)$. Injectivity depends on the domain: x^2 is not injective on $\mathbb{R}$ but is on $[0, \infty)$. So $x \mapsto x^2$ as a function $[0, \infty) \to [0, \infty)$ is a bijection, and its inverse is the square root, matching what we found earlier.

Worked example. Classify, with reasons:

1. $f : \mathbb{Z} \to \mathbb{Z},\ f(n) = 2n$. Injective: $2n_1 = 2n_2 \Rightarrow n_1 = n_2$. Not surjective: 3 is not twice any integer. Not bijective.
2. $g : \mathbb{R} \to \mathbb{R},\ g(x) = 2x$. Injective as before. Surjective: given any y, $x = y/2$ works. Bijective, with inverse $y \mapsto y/2$. The **same formula** changed its classification because the sets changed.
3. $h : \{0, 1, \ldots, 19\} \to \{0, 1, 2, 3, 4\},\ h(n) = n \bmod 5$. Surjective (0–4 are each hit, four times). Not injective (0 and 5 collide).

```python
def is_injective(f, domain):
    seen = {}
    for x in domain:
        y = f(x)
        if y in seen:
            return False, (seen[y], x)    # two inputs, one output
        seen[y] = x
    return True, None

def is_surjective(f, domain, codomain):
    missed = set(codomain) - {f(x) for x in domain}
    return not missed, missed

D = range(-3, 4)
print(is_injective(lambda x: x ** 2, D))
print(is_injective(lambda x: 2 * x + 1, D))
print(is_surjective(lambda x: x % 5, range(20), range(5)))
print(is_surjective(lambda x: 2 * x, range(5), range(10)))
```

```
(False, (-1, 1))
(True, None)
(True, set())
(False, {1, 3, 5, 7, 9})
```

On a **finite** domain these checks are complete proofs: the program examines every input. The injectivity check is the same algorithm as "detect duplicate values with a dictionary".

Computer-science connection: the -jectives in the wild

- **Injective:** primary keys (distinct rows must have distinct keys), lossless encodings (UTF-8 maps distinct code points to distinct byte strings), user-ID assignment. Hash functions are the famous **non**-injective case.
- **Surjective:** a load balancer that sends traffic to every server; a hash function onto n buckets that leaves no bucket permanently empty; a pseudo-random generator that can produce every value in its declared range.
- **Bijective:** permutations (shuffling a deck, reordering an array: a bijection from positions to positions), encryption on fixed-size blocks (AES is a bijection on 128-bit blocks, which is precisely why it can be decrypted), and format conversions that round-trip perfectly. Counting arguments often work by exhibiting a bijection between something hard to count and something easy (Volume 3, Chapter 24).



How big is infinity?

Georg Cantor (1870s) defined two sets to have the **same size** when there is a bijection between them. Surprising consequences follow. The even numbers are "as many" as all natural numbers (pair $n \leftrightarrow 2n$), even though they are a proper subset. The rationals can be listed in a single sequence, so they too match the naturals. But the real numbers cannot: Cantor's **diagonal argument** shows that any list of reals misses one. So there are different sizes of infinity, and the same diagonal argument, reused by Turing in 1936, shows that some problems (like deciding whether an arbitrary program halts) cannot be solved by any program.



Collision finder

Type keys into a hash table with an adjustable number of buckets and a choice of hash function (length of string, sum of character codes, CRC32). Watch the buckets fill, count collisions, and compare against the "birthday" prediction that collisions appear after roughly $\sqrt{n}$ keys in n buckets (Volume 3, Chapter 29). Then switch the codomain to as many buckets as keys and try to find a hash that is a bijection.

US curriculum

Pure versus impure functions, hash functions and lookup tables are computer-science topics (AP Computer Science A and Principles touch on methods and data structures). Injective, surjective and bijective are taught in university discrete mathematics; high schools use "one-to-one" (and occasionally "onto") in **Precalculus**.

11.12. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Graphing calculator
- Statistics

11.13. Exercises

1. **(Recognition)** Which of these are functions of x? (a) The relation $\{(1,2),(2,3),(3,2)\}$. (b) The relation $\{(1,2),(1,3),(2,4)\}$. (c) $y^2 = x$. (d) $y = x^3 - x$.
2. **(Recognition)** For $f:\mathbb{R}\to\mathbb{R},\ f(x)=x^2+1$, state the domain, codomain and range.
3. **(Application)** Let $f(x)=3x^2-x$. Find $f(-2)$ and simplify $f(x+h)-f(x)$.
4. **(Application)** Find the natural domain of (a) $\dfrac{x+1}{x^2-5x+6}$ and (b) $\sqrt{6-2x}$.
5. **(Application)** Describe the transformation from $y=\sqrt{x}$ to $y=3\sqrt{x+4}-2$, and give the image of the point $(9,3)$.
6. **(Application)** Let $f(x)=2x-1$ and $g(x)=x^2$. Find $(f\circ g)(x)$, $(g\circ f)(x)$ and $(f\circ f)(x)$.
7. **(Application)** Find the inverse of $f(x)=\dfrac{x}{x+1}$ and state its domain.
8. **(Application)** Classify as even, odd or neither: (a) x^5-2x^3, (b) $x^2+|x|$, (c) x^3+1.
9. **(Application)** Find the holes, vertical asymptotes and horizontal asymptote of $R(x)=\dfrac{3x^2-3x}{x^2-1}$.
10. **(Problem solving)** A phone plan costs \$20 per month including 5 GB of data, plus \$10 per extra GB **or part thereof**. Write the monthly cost $C(d)$ for data use d GB and find $C(7.2)$.
11. **(Problem solving)** For $f:\{0,1,\dots,9\}\to\{0,1,\dots,9\},\ f(n)=3n \bmod 10$, decide whether f is bijective and, if so, find f^{-1}. What changes if 3 is replaced by 4?
12. **(Problem solving)** Explain why `random.random()` is not a mathematical function, and describe how a library could provide a pure random-number generator.

▼ **Answers**

1. (a) Function: each first coordinate appears once (2 is hit twice, which is allowed). (b) Not a function: input 1 has two outputs. (c) Not a function of x: $x = 4$ gives $y = \pm 2$ (it fails the vertical line test). (d) Function.

2. Domain $\mathbb{R}$, codomain $\mathbb{R}$, range $[1, \infty)$ (since $x^2 \geq 0$, $x^2 + 1 \geq 1$, and every value ≥ 1 is reached at $x = \sqrt{y - 1}$).

3. $f(-2) = 3(4) + 2 = 14$.
   $f(x + h) = 3(x^2 + 2xh + h^2) - x - h = 3x^2 + 6xh + 3h^2 - x - h$, so
   $f(x + h) - f(x) = 6xh + 3h^2 - h$. (Dividing by h gives $6x + 3h - 1$; this **difference quotient** reappears in Volume 2, Chapter 17 .)

4. (a) $x^2 - 5x + 6 = (x - 2)(x - 3)$, so domain $\mathbb{R} \setminus \{2, 3\}$. (b) $6 - 2x \geq 0 \iff x \leq 3$: domain $(-\infty, 3]$.

5. $a = 3, b = 1, h = -4, k = -2$: shift left 4, stretch vertically by 3, shift down 2.
   $(9, 3) \mapsto (9 - 4,\ 3 \cdot 3 - 2) = (5, 7)$. Check: $3\sqrt{5 + 4} - 2 = 9 - 2 = 7$.

6. $(f \circ g)(x) = 2x^2 - 1$; $(g \circ f)(x) = (2x - 1)^2 = 4x^2 - 4x + 1$;
   $(f \circ f)(x) = 2(2x - 1) - 1 = 4x - 3$.

7. $y = x/(x + 1) \Rightarrow y(x + 1) = x \Rightarrow xy + y = x \Rightarrow y = x - xy = x(1 - y) \Rightarrow x = y/(1 - y)$. So $f^{-1}(x) = \dfrac{x}{1 - x}$, domain $x \neq 1$ (1 is the horizontal asymptote of f, never reached). Check: $f(1) = 1/2$, $f^{-1}(1/2) = (1/2)/(1/2) = 1$.

8. (a) Odd: $(-x)^5 - 2(-x)^3 = -x^5 + 2x^3 = -(x^5 - 2x^3)$. (b) Even: $(-x)^2 + |-x| = x^2 + |x|$. (c) Neither: $f(1) = 2$, $f(-1) = 0$, which is neither 2 nor -2.

9. $R(x) = \dfrac{3x(x - 1)}{(x - 1)(x + 1)}$. Common factor $x - 1$: hole at $x = 1$, height $3(1)/(1 + 1) = 3/2$, so the hole is at $(1, 3/2)$. Remaining denominator zero: vertical asymptote $x = -1$. Equal degrees, leading coefficients 3 and 1: horizontal asymptote $y = 3$.

10. $C(d) = 20$ for $0 \leq d \leq 5$; $C(d) = 20 + 10\lceil d - 5 \rceil$ for $d > 5$. (The single formula $20 + 10 \max(0, \lceil d - 5 \rceil)$ also works.) $C(7.2) = 20 + 10\lceil 2.2 \rceil = 20 + 30 = 50$ dollars.

11. Outputs for $n = 0, \ldots, 9$: 0, 3, 6, 9, 2, 5, 8, 1, 4, 7, all different and covering 0–9, so f is a bijection. Reading the table backwards, $f^{-1}(m) = 7m \bmod 10$ (because $3 \times 7 = 21 \equiv 1 \pmod{10}$; check $f^{-1}(3) = 21 \bmod 10 = 1$). With 4: outputs 0, 4, 8, 2, 6, 0, … only even numbers, so neither injective nor surjective. The difference is that 3 shares no factor with 10 while 4 does (Volume 3, Chapter 23).

12. Calling it twice with the same (empty) input gives different outputs, so it depends on a hidden input: the generator's internal state, which each call also changes (a side effect). A pure design passes the state explicitly: `next_value, new_state = step(state)`. NumPy's `Generator` objects and JAX's explicit random keys follow this idea.

11.14. Mastery check

1. State the definition of a function in one sentence, and explain why $x^2 + y^2 = 25$ does not define y as a function of x.
2. Find the natural domain of $f(x) = \dfrac{\sqrt{x+2}}{x^2 - 9}$ in interval notation.
3. Sketch $g(x) = -|x - 1| + 3$ by transforming $|x|$; give the vertex and the x-intercepts.
4. Write $h(x) = \sqrt{5 - x^3}$ as a composition $f \circ g$ of two simpler functions, and find its domain.
5. Find the inverse of $f(x) = \dfrac{4x - 1}{3}$ and verify it with one number.
6. Is $f(x) = x^3 - 4x$ even, odd or neither? Prove it.
7. Give the asymptotes of $R(x) = \dfrac{x + 5}{2x - 4}$.
8. For $f : \{a, b, c, d\} \to \{1, 2, 3\}$, explain why f cannot be injective, and say what that implies for hash functions.

Decision rule: All correct without hesitation → move on to Chapter 12. Stuck on 1–2 → reread the relevant section (Section 11.2 for 2, Section 11.5 for 3, Section 11.7 for 5). Stuck on more → work the chapter exercises first.

▼ **Mastery check answers**

1. A function from A to B assigns to each element of A exactly one element of B. For $x = 3$, both $y = 4$ and $y = -4$ satisfy the equation, so $x = 3$ would have two outputs (the circle fails the vertical line test).
2. Root: $x + 2 \geq 0 \iff x \geq -2$. Denominator: $x^2 - 9 \neq 0 \iff x \neq \pm 3$; -3 is already excluded. Domain: $[-2, 3) \cup (3, \infty)$.
3. $a = -1, h = 1, k = 3$: the V flipped upside down, corner (vertex) at $(1, 3)$. x-intercepts: $|x - 1| = 3 \iff x - 1 = \pm 3 \iff x = 4$ or $x = -2$.
4. Inner $g(x) = 5 - x^3$, outer $f(u) = \sqrt{u}$. Domain: $5 - x^3 \geq 0 \iff x^3 \leq 5 \iff x \leq \sqrt[3]{5}$, i.e. $(-\infty, \sqrt[3]{5}]$ (cubing is increasing, so the inequality direction is preserved).
5. $y = (4x - 1)/3 \Rightarrow 3y = 4x - 1 \Rightarrow 3y + 1 = 4x \Rightarrow x = (3y + 1)/4$. So $f^{-1}(x) = (3x + 1)/4$. Check: $f(1) = 1$, $f^{-1}(1) = 1$; better, $f(4) = 5$, $f^{-1}(5) = 16/4 = 4$.
6. Odd: $f(-x) = (-x)^3 - 4(-x) = -x^3 + 4x = -(x^3 - 4x) = -f(x)$ for every x.
7. Vertical: $2x - 4 = 0 \iff x = 2$ (numerator $x + 5$ is not zero there, so no hole). Horizontal: equal degrees, leading coefficients 1 and 2, so $y = 1/2$.
8. Four inputs and three outputs: by the pigeonhole principle two inputs must share an output. Any hash function from a larger set of keys to a smaller set of buckets must have collisions; good hashing makes them rare and well spread, never impossible.

Chapter 12. Exponentials and Logarithms

Some quantities grow by adding the same amount each step. Others grow by multiplying by the same factor each step, and those behave in a completely different way. This chapter is about the second kind. It covers exponential functions, the special number e that comes out of compound interest, and the logarithm, which answers the question "what power?". Logarithms turn multiplication into addition, which is why they show up in decibels, earthquake magnitudes, binary search, information theory and the numerical guts of machine learning.

Why it matters: savings, loans, inflation, epidemics, radioactive dating, drug doses, cooling coffee, network retry timers and the cost of brute-forcing a password all follow the same multiply-by-a-constant rule. Misjudging exponential growth is one of the most expensive mistakes people make with money and with forecasts, and logarithms are the tool that tames it.

Where this chapter sits

You need: exponent laws, including zero, negative and fractional exponents (Chapter 3), solving linear equations (Chapter 5), and the idea of a function, its graph and its inverse (Chapter 11).

This chapter unlocks: sequences and series, especially geometric ones (Chapter 14); the limit that defines e (Volume 2, Chapter 16); the derivative of e^x and $\ln x$ (Volume 2, Chapter 17); growth-rate arguments in algorithm analysis (Volume 3, Chapter 24 , Volume 3, Chapter 25); probability distributions and log-likelihood (Volume 3, Chapter 30 , Volume 3, Chapter 31); floating-point and numerical stability (Volume 2, Chapter 32).

If you are in a hurry: the essential sections are Section 12.1, Section 12.3, Section 12.4 and Section 12.5. The CS reader should not skip Section 12.9.

12.1. Growth by multiplication

12.1.1. The problem

Take a sheet of paper 0.1 mm thick and fold it in half. It is now 0.2 mm thick. Fold it again: 0.4 mm. How many folds would it take for the stack to reach the Moon, about 384,400 km away?

Most people guess thousands, or millions. The answer is **42**.

Here is a second puzzle that is really the same one. A rumour starts with one person. Every hour, everyone who knows it tells one new person. After one day (24 hours), how many people know?

$$2^{24} = 16{,}777{,}216.$$

After a day and a half, more people than live on Earth would "know", which is obviously impossible. Something has to give. We come back to that in Section 12.7.

What these puzzles share is a quantity that gets **multiplied by the same factor** at every step. That is the defining feature of exponential behaviour, and human intuition is very bad at it.

12.1.2. Intuition: constant difference versus constant ratio

Compare two savings plans, each starting at $100:

- Plan L adds $10 every year.
- Plan E adds 10% of the current balance every year, so the balance is multiplied by 1.1.

Year	Plan L	L: change	Plan E	E: ratio
0	100	—	100.00	—
1	110	+10	110.00	×1.1
2	120	+10	121.00	×1.1
3	130	+10	133.10	×1.1
10	200	+10 each	259.37	×1.1 each
50	600	+10 each	11,739.09	×1.1 each

In the early years the two plans are almost the same. Over time the multiplying plan leaves the adding plan far behind. The rule of thumb:

- **Linear** growth: equal steps in input give equal **differences** in output. The graph is a straight line (Chapter 5).
- **Exponential** growth: equal steps in input give equal **ratios** in output. The graph bends upward faster and faster.

You can recognise exponential data in a table by dividing each value by the one before. If the ratios are all (roughly) the same, the data is exponential.

12.1.3. Vocabulary and notation

Exponential function

An **exponential function** has the variable in the **exponent**:

$$f(x) = a \cdot b^x, \qquad b > 0,\ b \neq 1.$$

- b is the **base** or **growth factor**: the number you multiply by for each unit step in x.
- a is the **initial value**, the output when $x = 0$, because $b^0 = 1$.

Compare it with a **power function** such as x^2, where the variable is in the base. x^2 and 2^x are very different functions.

Growth rate, decay rate

If something grows by a percentage r per step (written as a decimal, so 5% means $r = 0.05$), its growth factor is $b = 1 + r$. If it **decays** (shrinks) by r per step, then $b = 1 - r$.

- $b > 1$: **exponential growth**.
- $0 < b < 1$: **exponential decay**.

r is the **growth rate** (or decay rate). b is the **growth factor**. Don't mix the two up: a 5% rate is a factor of 1.05.

Doubling time and half-life

The **doubling time** of an exponentially growing quantity is the time it takes to double. The **half-life** of a decaying quantity is the time it takes to halve. For an exponential process these are **constant**: going from 100 to 200 takes exactly as long as going from 1,000 to 2,000.

With a half-life T, the amount after time t is

$$N(t) = N_0\left(\tfrac{1}{2}\right)^{t/T}.$$

Here N_0 ("N nought") is the starting amount and t/T counts how many half-lives have gone by.

12.1.4. Why the model works

If a quantity is multiplied by b each step and starts at a, then

$$
\begin{aligned}
f(0) &= a \\
f(1) &= ab \\
f(2) &= (ab)b = ab^2 \\
f(3) &= (ab^2)b = ab^3 \\
&\vdots \\
f(n) &= ab^n.
\end{aligned}
$$

So "multiply by b every step" turns into "raise b to the number of steps". The exponent counts the multiplications.

Now look again at the folding puzzle. Thickness after n folds is 0.1×2^n mm. For $n = 42$:

$$0.1 \times 2^{42}\ \text{mm} = 0.1 \times 4{,}398{,}046{,}511{,}104\ \text{mm} \approx 4.40 \times 10^{11}\ \text{mm} \approx 439{,}800\ \text{km},$$

which is more than the 384,400 km to the Moon. (41 folds gives about 219,900 km, which is not enough.) Doubling 42 times multiplies by more than four trillion.

From Chapter 3 you know that b^x makes sense for any real x, not only whole numbers: $b^{1/2} = \sqrt{b}$, $b^{-1} = 1/b$, and irrational exponents are filled in by continuity. So $f(x) = ab^x$ is defined on the whole real line and gives a smooth curve.

12.1.5. The graph of b^x

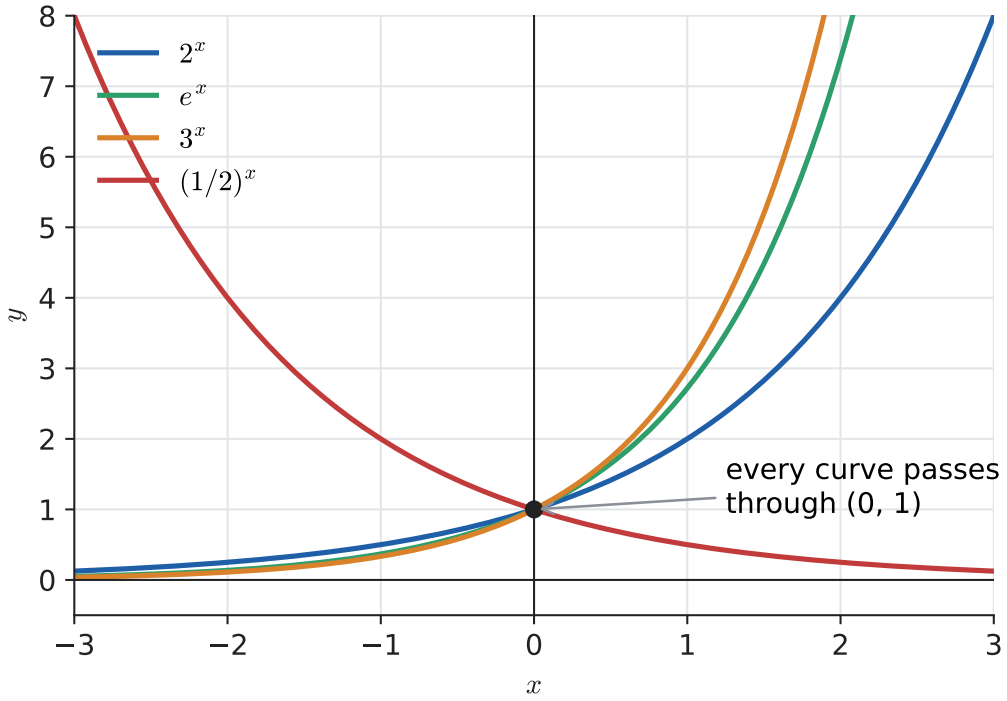


Figure 92. Exponential functions with several bases. Every graph passes through $(0, 1)$. Bases above 1 rise; bases between 0 and 1 fall.

Every graph of $y = b^x$ with $b > 0$, $b \neq 1$ shares these features:

- **Domain:** all real numbers. You can raise a positive base to any power.

- **Range:** $y > 0$. A positive base raised to any power is positive, so the graph never touches or crosses the x-axis.
- **y-intercept:** $(0, 1)$, because $b^0 = 1$.
- **Horizontal asymptote:** the x-axis, $y = 0$. For $b > 1$ the curve gets arbitrarily close to it as $x \to -\infty$. For $b < 1$ that happens as $x \to +\infty$.
- **One-to-one:** each output comes from exactly one input. That means the function has an inverse, and the inverse is the logarithm.
- $(1/b)^x = b^{-x}$, so the graph of $(1/2)^x$ is the graph of 2^x reflected in the y-axis.



Asymptote

An **asymptote** is a line that a curve approaches more and more closely without reaching it (at least far out). $y = 0$ is a **horizontal asymptote** of $y = 2^x$: as x becomes very negative, 2^x gets as close to 0 as you like but stays positive.

12.1.6. Worked example: radioactive decay

Carbon-14 has a half-life of about 5,730 years. A sample starts with 200 micrograms of carbon-14. How much is left after 20,000 years?

$$
\begin{aligned}
N(t) &= N_0\left(\tfrac{1}{2}\right)^{t/T} && \text{(half-life model)} \\
N(20000) &= 200\left(\tfrac{1}{2}\right)^{20000/5730} && \text{(substitute)} \\
&= 200\left(\tfrac{1}{2}\right)^{3.4904} && \text{(about 3.49 half-lives)} \\
&= 200 \times 0.08898 && \text{(evaluate } 0.5^{3.4904}\text{)} \\
&\approx 17.8 \ \mu\text{g}.
\end{aligned}
$$

Sanity check: after 3 half-lives you would have $200/8 = 25$, and after 4 you would have $200/16 = 12.5$. 17.8 falls between them, as it should.

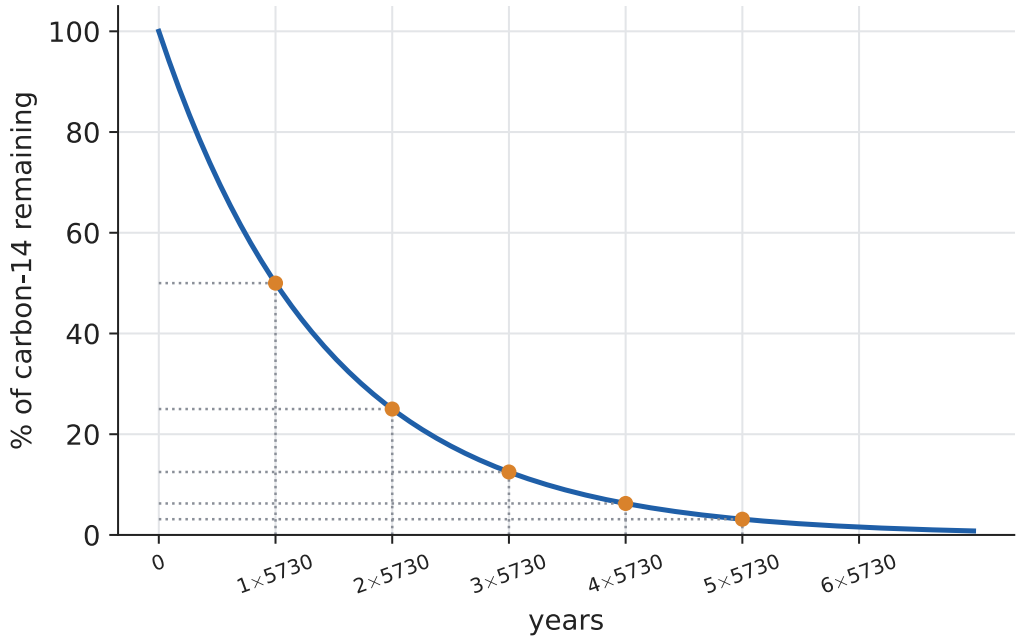


Figure 93. Carbon-14 halving every 5,730 years. Each orange dot is one more half-life.

The **reverse** question, "how old is a sample with 30% of its carbon-14 left?", needs the exponent as the unknown. We can't answer it with what we have yet. That is exactly what the logarithm is for, and we solve it in Section 12.5.



Rate versus factor, and percentage points

- "Grows 5% a year" means multiply by $\mathbf{1.05}$, not by 0.05 and not by 5.
- "Falls 20% then rises 20%" does **not** bring you back to where you started: $\mathbf{0.8 \times 1.2 = 0.96}$, a 4% loss.
- A model with $b < 0$, such as $(-2)^x$, is not an exponential function. It flips sign at each step and is undefined at $x = 1/2$.
- $a \cdot b^x$ is not $(ab)^x$. The exponent applies only to b (order of operations, Chapter 4).

12.1.7. Where you'd use this

- **Money.** Inflation of 3% a year multiplies prices by $1.03^{20} \approx 1.81$ over 20 years: a $5 sandwich becomes about $9. The same arithmetic grows a pension pot, and works against you on unpaid debt.
- **Medicine.** Caffeine has a half-life of about 5 hours in a typical adult. A 200 mg coffee at 8 a.m. leaves $200 \times (\frac{1}{2})^{15/5} = 200/8 = 25$ mg in your blood at 11 p.m., about the same as a can of cola. Doctors space doses of many drugs using exactly this half-life model.
- **Computing: exponential backoff.** When a network request fails, well-behaved clients wait 1 s, then 2 s, 4 s, 8 s, … before retrying, so a struggling server isn't hammered. The tenth retry waits $2^9 = 512$ s, over 8 minutes, which is why real implementations cap the wait.

- **Security.** Every extra character multiplies the number of possible passwords. With 62 letters and digits, 8 characters give $62^8 \approx 2.2 \times 10^{14}$ possibilities, about 6 hours of work for an attacker testing 10^{10} guesses a second; 12 characters give $62^{12} \approx 3.2 \times 10^{21}$, about 10,000 years. Length beats cleverness because length is in the exponent.



> *The king, the chessboard and the rice*
>
> An old legend has the inventor of chess ask for a reward of one grain of rice on the first square, two on the second, four on the third, and so on for all 64 squares. The total is $2^{64} - 1 \approx 1.8 \times 10^{19}$ grains, hundreds of years of current world rice production. You will recognise $2^{64} - 1$ as the largest unsigned 64-bit integer, `UINT64_MAX`.

12.2. Compound interest and the number e

12.2.1. The problem

A bank offers 100% interest a year. (Not a real bank.) Put in $1 and after a year you have $2. A rival bank says: "We pay the same rate, but we compound **twice** a year: 50% after six months, then 50% on the new balance." Another compounds monthly. Another daily. Another every second.

Does compounding more often make you arbitrarily rich, or is there a ceiling?

12.2.2. Intuition

Compounding more often helps, because interest starts earning interest sooner. But each split also makes the individual pieces smaller. These two effects pull against each other, and it isn't obvious which wins.

12.2.3. The calculation

Split the year into n periods. Each period pays $1/n$ of the annual rate, so the balance is multiplied by $1 + 1/n$ n times:

$$\text{balance after one year} = \left(1 + \frac{1}{n}\right)^n.$$

Compounding	n	$(1 + 1/n)^n$
yearly	1	2
half-yearly	2	2.25
quarterly	4	2.441406…
monthly	12	2.613035…
weekly	52	2.692597…
daily	365	2.714567…
hourly	8,760	2.718127…
every minute	525,600	2.718279…

The balance grows, but more and more slowly, and it levels off just above 2.718. That ceiling is the number

$$e = 2.718281828459045\ldots$$

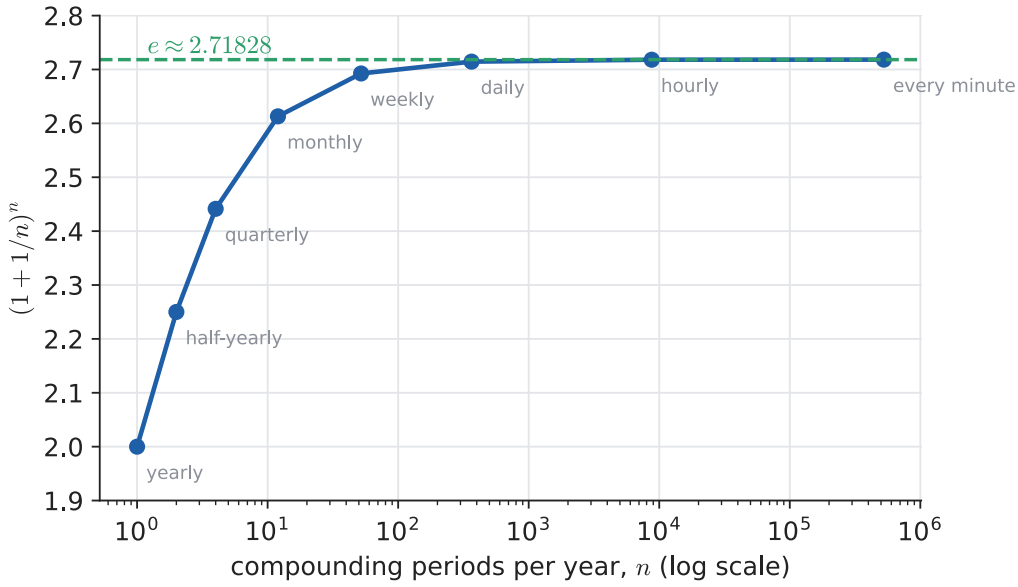


Figure 94. Daily, hourly and every-minute compounding all give practically the same result. They are converging on e.

The formal definition uses a **limit**, which is the subject of Volume 2, Chapter 16 :

$$e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n.$$

Read "lim as $n \to \infty$" as "the value this expression settles towards as n grows without bound".



The number e

$e \approx 2.71828$ is **Euler's number**. It is irrational (its decimal expansion never repeats), like π. Two key facts about it:

- $e = \lim_{n \to \infty} (1 + 1/n)^n$, the limit of continuous compounding.
- The function e^x is **its own rate of change**. At every point, the slope of the graph of $y = e^x$ equals its height. That is why e^x is called **the** exponential function, and why it is written `exp(x)` in code.

12.2.4. Continuous compounding in general

Suppose the annual rate is r instead of 100%, you invest a **principal** P for t years, and you compound n times a year. The balance is

$$A = P\left(1 + \frac{r}{n}\right)^{nt}.$$

Let $m = n/r$. Then $r/n = 1/m$ and $nt = m \cdot rt$, so

$$\begin{aligned} A &= P\left(1 + \frac{1}{m}\right)^{m \cdot rt} \\ &= P\left[\left(1 + \frac{1}{m}\right)^{m}\right]^{rt} && \text{(power of a power: } x^{ab} = (x^a)^b\text{)} \\ &\to P\,e^{rt} \quad \text{as } n \text{ (and so } m) \to \infty. \end{aligned}$$

That gives the **continuous compounding** formula

$$A = Pe^{rt}.$$

12.2.5. Worked example: three ways to compound

\$1,000 at 5% a year for 10 years:

$$\begin{aligned} \textbf{Annually:} \quad & A = 1000(1.05)^{10} = 1000 \times 1.628895 = \$1{,}628.89 \\ \textbf{Monthly:} \quad & A = 1000\left(1 + \tfrac{0.05}{12}\right)^{120} = 1000 \times 1.647009 = \$1{,}647.01 \\ \textbf{Continuously:} \quad & A = 1000\,e^{0.05 \times 10} = 1000\,e^{0.5} = 1000 \times 1.648721 = \$1{,}648.72 \end{aligned}$$

Going from annual to monthly compounding earns \$18. Going from monthly to continuous earns only \$1.71 more. That matches the levelling off in the table.

12.2.6. Seeing the limit numerically

```python
import math

for n in [1, 10, 100, 10_000, 1_000_000, 10**8, 10**12, 10**16]:
    print(f"{n:>22,}  {(1 + 1/n)**n:.15f}")
print(f"{'math.e':>22}  {math.e:.15f}")
```

```
                    1  2.000000000000000
                   10  2.593742460100002
                  100  2.704813829421528
               10,000  2.718145926824926
            1,000,000  2.718280469095753
          100,000,000  2.718281798347358
    1,000,000,000,000  2.718523496037238
10,000,000,000,000,000  1.000000000000000
                math.e  2.718281828459045
```

The values close in on e up to $n = 10^8$. Then they go **wrong**: 10^{12} is worse, and at 10^{16} the answer is exactly 1. The reason is floating point. `1 + 1/n` is rounded to the nearest representable double, and when $1/n$ is below about 1.1×10^{-16}, `1 + 1/n` rounds to exactly `1.0`. We return to this kind of failure in Volume 2, Chapter 32 . The general lesson is that a numerical approach to a limit has a sweet spot. Push too far and rounding error takes over.



APR is not APY

Banks quote a **nominal** annual rate (APR) and an **effective** annual yield (APY). 12% APR compounded monthly gives $(1.01)^{12} - 1 = 12.68\%$ APY. If you compare loans by APR when they compound at different frequencies, you are comparing different things.

12.2.7. Where you'd use this

Continuous compounding is how many financial models price bonds and options, but e turns up far beyond banking, wherever many small independent chances or a growth rate proportional to size are involved:

- **Biology:** a bacterial population with continuous growth rate $k = 0.7$ per hour is $N_0 e^{0.7t}$; after 5 hours it is $e^{3.5} \approx 33$ times larger.
- **Physics and electronics:** a capacitor discharging through a resistor keeps the fraction $e^{-t/RC}$ of its voltage. After one **time constant** $t = RC$, $e^{-1} \approx 37\%$ remains; after five, under 1%, which is why engineers treat "5 RC" as fully discharged.
- **Computing:** throw n keys at random into a hash table with n buckets. Each bucket stays empty with probability $(1 - 1/n)^n \approx 1/e \approx 0.368$, so about 37% of the buckets are empty and the rest hold everything. The same limit that defined e predicts how unevenly random hashing fills a table.



Who found e?

Jacob Bernoulli ran into the compound-interest limit in 1683 while studying exactly this question. He showed it was between 2 and 3. Leonhard Euler later gave it the letter e (probably not after himself, despite the popular story), computed it to more than 20 decimal places and proved it irrational. The same number turns up in probability: if you shuffle n letters into n envelopes at random, the chance that **none** lands in the right envelope approaches $1/e \approx 0.368$.

12.3. The logarithm: "what power?"

12.3.1. The problem

From Section 12.1 you can answer "2^{10} is what?" (1,024). Now turn the question round:

> **$2^{?} = 1024$. *What power* *of 2 gives 1,024?***

That is the question a logarithm answers. The answer is 10, and we write

$$\log_2 1024 = 10.$$

Questions like this come up constantly. How many years until my investment doubles? How old is this fossil? How many times can I halve a list of a million items before one item is left? How many bits does it take to count to a billion? Each asks for an unknown **exponent**.

12.3.2. Definition



Logarithm

For a base $b > 0$, $b \neq 1$, and a positive number x:

$$\log_b x = y \quad \Longleftrightarrow \quad b^y = x.$$

Read $\log_b x$ as "log base b of x". It is **the exponent you put on b to get x**. In the equation $b^y = x$:

- b is the **base**,
- y is the **logarithm** (the exponent),
- x is sometimes called the **argument** of the log.

The double arrow $\Longleftrightarrow$ means "if and only if": each statement implies the other, so they are two ways of saying the same thing.

The word to hold onto: **a logarithm is an exponent.** Every fact about logarithms is a fact about exponents written the other way round.

12.3.3. Worked examples: evaluating logs by hand

Each of these is answered by asking "what power?":

$$\begin{aligned}
\log_2 8 &= 3 && \text{because } 2^3 = 8 \\
\log_{10} 1000 &= 3 && \text{because } 10^3 = 1000 \\
\log_5 1 &= 0 && \text{because } 5^0 = 1 \\
\log_3 \tfrac{1}{9} &= -2 && \text{because } 3^{-2} = \tfrac{1}{3^2} = \tfrac{1}{9} \\
\log_4 2 &= \tfrac{1}{2} && \text{because } 4^{1/2} = \sqrt{4} = 2 \\
\log_{10} 0.001 &= -3 && \text{because } 10^{-3} = 0.001 \\
\log_b b &= 1 && \text{because } b^1 = b
\end{aligned}$$

One more, a bit harder: $\log_8 32$. Both 8 and 32 are powers of 2: $8 = 2^3$ and $32 = 2^5$. We want y with

$$\begin{aligned} 8^y &= 32 \\ (2^3)^y &= 2^5 && \text{(rewrite with base 2)} \\ 2^{3y} &= 2^5 && \text{(power of a power)} \\ 3y &= 5 && \text{(same base, so the exponents must match)} \\ y &= \tfrac{5}{3}. \end{aligned}$$

So $\log_8 32 = 5/3$. Check: $8^{5/3} = (\sqrt[3]{8})^5 = 2^5 = 32$. Correct.

12.3.4. The log function and its graph

Because $y = b^x$ is one-to-one, it has an inverse function (Chapter 11), and that inverse is $y = \log_b x$. Inverse functions undo each other:

$$\log_b(b^x) = x \quad \text{for all real } x, \qquad b^{\log_b x} = x \quad \text{for all } x > 0.$$

The graph of an inverse function is the original graph **reflected in the line $y = x$**. Every point (p, q) on $y = 2^x$ becomes the point (q, p) on $y = \log_2 x$.

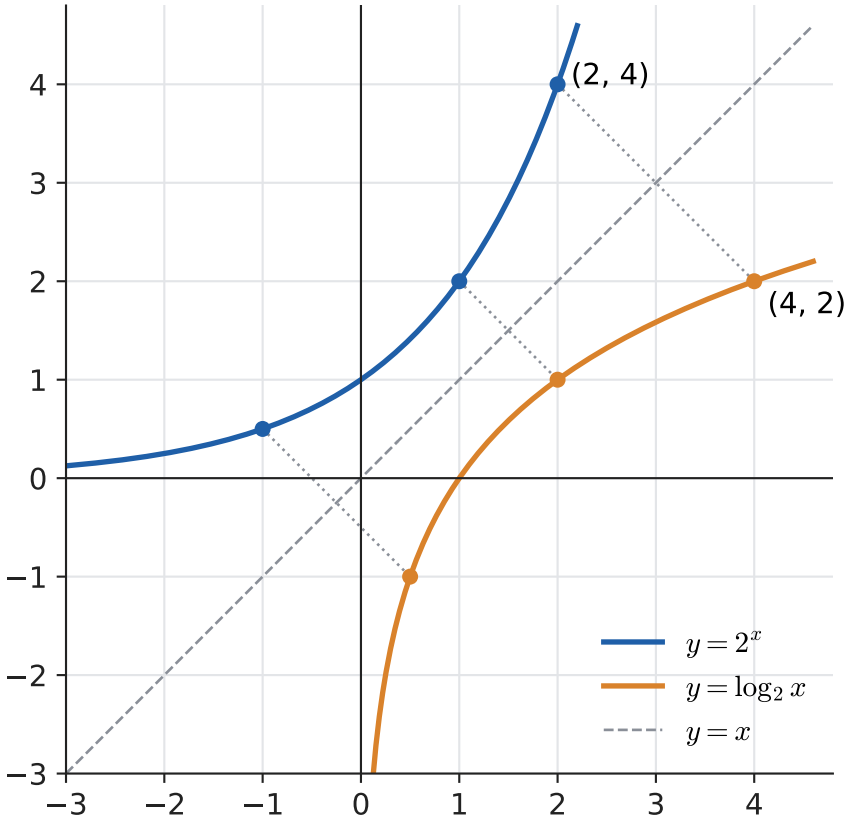


Figure 95. $y = 2^x$ and its inverse $y = \log_2 x$, mirror images in the line $y = x$. The point $(2, 4)$ on one matches $(4, 2)$ on the other.

Swapping the roles of x and y in the exponential's properties gives the logarithm's properties:

	$y = b^x$ $(b > 1)$	$y = \log_b x$ $(b > 1)$
Domain	all reals	$x > 0$ only
Range	$y > 0$	all reals
Key point	$(0, 1)$	$(1, 0)$
Asymptote	horizontal, $y = 0$	vertical, $x = 0$
Growth	faster and faster	slower and slower, but never stops

The last row matters a lot in computing. $\log_2 x$ does keep increasing without limit, but **extremely** slowly. $\log_2$ of a thousand is about 10, of a million about 20, of a billion about 30. Multiplying the input by 1,000 adds only about 10 to the output. That is why an algorithm that takes $\log n$ steps scales so well.



Logs of zero and negative numbers do not exist (in the reals)

$\log_b x$ is defined only for $x > 0$, because a positive base raised to any real power is positive. $\log_{10}(-100)$ has no real answer, and $\log 0$ is undefined. (As $x \to 0^+$ the log heads to $-\infty$.) In code, `math.log(0)` raises `ValueError: math domain error`, while `numpy.log(0)` returns `-inf` with a warning. Either way, a zero probability or zero count sitting in a log is a common bug.

12.4. The laws of logarithms, derived

12.4.1. The problem

Before calculators, multiplying two seven-digit numbers by hand was slow and easy to get wrong, while adding them was fast. Navigators, astronomers and engineers wanted a way to **turn multiplication into addition**. Logarithms did that, and the reason is a single exponent law.

12.4.2. Why it works: the derivations

The log laws aren't new rules to memorise. Each is an exponent law from Chapter 3 read backwards. In everything below, $b > 0$, $b \neq 1$, and $M, N > 0$.

Product law. Let $m = \log_b M$ and $n = \log_b N$. By definition, $b^m = M$ and $b^n = N$. Then

$$\begin{aligned} MN &= b^m \cdot b^n \\ &= b^{m+n} && \text{(exponent law: } b^m b^n = b^{m+n}\text{)} \\ \log_b(MN) &= m + n && \text{(definition of log: the exponent on } b\text{)} \\ &= \log_b M + \log_b N. \end{aligned}$$

Quotient law. Same setup:

$$\frac{M}{N} = \frac{b^m}{b^n} = b^{m-n} \qquad \text{(exponent law: } b^m/b^n = b^{m-n}\text{)}$$
$$\log_b\left(\frac{M}{N}\right) = m - n = \log_b M - \log_b N.$$

Power law. For any real k:

$$M^k = (b^m)^k = b^{mk} \qquad \text{(exponent law: } (b^m)^k = b^{mk}\text{)}$$
$$\log_b\left(M^k\right) = mk = k \log_b M.$$

Change of base. Suppose you need $\log_2 50$ but your calculator only has $\log_{10}$ and $\ln$. Let $y = \log_b x$, so $b^y = x$. Take log base c (any convenient base) of both sides:

$$\begin{aligned} b^y &= x \\ \log_c(b^y) &= \log_c x \qquad && \text{(apply } \log_c \text{ to both sides)} \\ y \log_c b &= \log_c x \qquad && \text{(power law)} \\ y &= \frac{\log_c x}{\log_c b} \qquad && \text{(divide; } \log_c b \neq 0 \text{ since } b \neq 1\text{).} \end{aligned}$$

So

$$\log_b x = \frac{\log_c x}{\log_c b}.$$

In words: **every logarithm is a constant multiple of every other**.
$\log_2 x = \log_{10} x / \log_{10} 2 \approx 3.3219 \log_{10} x$. Changing the base just rescales the vertical axis of the graph.

The laws of logarithms

For $b, c > 0, b, c \neq 1, M, N > 0$, and any real k:

Law	Logarithm form	Exponent law it comes from
Product	$\log_b(MN) = \log_b M + \log_b N$	$b^m b^n = b^{m+n}$
Quotient	$\log_b(M/N) = \log_b M - \log_b N$	$b^m / b^n = b^{m-n}$
Power	$\log_b(M^k) = k \log_b M$	$(b^m)^k = b^{mk}$
Zero	$\log_b 1 = 0$	$b^0 = 1$
Identity	$\log_b b = 1$	$b^1 = b$
Inverse	$\log_b(b^x) = x, b^{\log_b x} = x$	definition
Change of base	$\log_b x = \dfrac{\log_c x}{\log_c b}$	power law

12.4.3. Common, natural and binary logarithms

Three bases account for nearly every logarithm you will meet.

Common, natural and binary logarithms

- **Common log**, base 10: $\log_{10} x$, often written $\log x$ in science and engineering and on calculators. It counts decimal digits: $\log_{10} 1000 = 3$.
- **Natural log**, base e: $\ln x = \log_e x$ ("ell-en" or "lawn"). It is the one that makes calculus clean, so mathematicians often write plain $\log x$ to mean $\ln x$.
- **Binary log**, base 2: $\log_2 x$, sometimes written $\lg x$. It counts bits and halvings, and in algorithm analysis an unadorned $\log n$ usually means $\log_2 n$.

In code: Python's `math.log(x)` is the **natural** log, and `math.log10`, `math.log2` and `math.log(x, b)` give the others. The same holds for C's `log`, JavaScript's `Math.log`, NumPy's `np.log` and Excel's `LN`. (Excel's `LOG` is base 10.) If a bare "log" shows up, check which base is meant.

Some values worth recognising:

$$\log_{10} 2 \approx 0.30103, \qquad \ln 2 \approx 0.693147, \qquad \ln 10 \approx 2.302585, \qquad \log_2 10 \approx 3.321928.$$

$\log_{10} 2 \approx 0.301$ says $2^{10} = 1024 \approx 10^3$: ten doublings add about three decimal digits. That is the familiar kilo/kibi relationship. Big-O notation doesn't care about the base of a log, because by change of base $\log_2 n$ and $\log_{10} n$ differ only by a constant factor.

12.4.4. Where you'd use this

- **Counting digits.** The number of decimal digits of a positive integer N is $\lfloor \log_{10} N \rfloor + 1$. How many digits does 2^{100} have? $\log_{10} 2^{100} = 100 \log_{10} 2 \approx 30.10$, so 31 digits, without computing the number.
- **Music.** Pitch is logarithmic: each octave doubles the frequency, and each of the 12 semitones in an octave multiplies it by $2^{1/12} \approx 1.0595$. The interval from 440 Hz (concert A) to 660 Hz is $12 \log_2(660/440) = 12 \log_2 1.5 \approx 7.02$ semitones: a perfect fifth.
- **Photography.** Each "stop" of exposure doubles or halves the light, so camera settings are a base-2 log scale. Going from f/4 to f/8 doubles the aperture's f-number, which quarters the aperture's area: two stops darker.
- **Computing.** $\log_2$ counts bits and halvings (see Section 12.9); the product law is why adding log-probabilities replaces multiplying tiny probabilities in spam filters and language models.

12.4.5. The method: expanding and condensing

Example 69. Procedure — Rewriting a logarithmic expression

Input: an expression involving logs of products, quotients and powers (to **expand**), or a sum and difference of logs with coefficients (to **condense**).

Steps (expanding):

1. Rewrite roots as fractional powers: $\sqrt[n]{u} = u^{1/n}$.
2. Split products into sums and quotients into differences (product and quotient laws).
3. Bring exponents down as coefficients (power law).
4. Evaluate any logs of known powers of the base ($\log_b b^k = k$).

Steps (condensing): run the steps backwards. First move coefficients up as exponents, then combine sums into products and differences into quotients, and finish with a **single** log.

Output: an equivalent expression. Check by plugging in a number on both sides.

12.4.6. Worked examples

Expand $\log_2\left(\dfrac{8x^3}{\sqrt{y}}\right)$:

$$\begin{aligned}
\log_2\left(\frac{8x^3}{\sqrt{y}}\right) &= \log_2\left(\frac{8x^3}{y^{1/2}}\right) && \text{(root as power)}\\
&= \log_2(8x^3) - \log_2(y^{1/2}) && \text{(quotient law)}\\
&= \log_2 8 + \log_2 x^3 - \log_2 y^{1/2} && \text{(product law)}\\
&= 3 + 3\log_2 x - \tfrac{1}{2}\log_2 y && (\log_2 8 = 3\text{; power law}).
\end{aligned}$$

Condense $2\ln a + \ln b - 3\ln c$ to a single log:

$$\begin{aligned}
2\ln a + \ln b - 3\ln c &= \ln a^2 + \ln b - \ln c^3 && \text{(power law, backwards)}\\
&= \ln(a^2 b) - \ln c^3 && \text{(product law, backwards)}\\
&= \ln\left(\frac{a^2 b}{c^3}\right) && \text{(quotient law, backwards)}.
\end{aligned}$$

Change of base: compute $\log_3 20$ with a calculator that only has $\ln$:

$$\log_3 20 = \frac{\ln 20}{\ln 3} = \frac{2.995732}{1.098612} \approx 2.726833.$$

Check: $3^{2.726833}$ should be 20. $3^2 = 9$ and $3^3 = 27$, so an exponent between 2 and 3 that lands closer to 3 is about right.



Laws that do not exist

The commonest log mistakes are made-up "laws" that look plausible:

- $\log(M+N) \neq \log M + \log N$. There is **no** simple rule for the log of a sum. (The product law has the product **inside**.)
- $\log(M-N) \neq \log M - \log N$.
- $\dfrac{\log M}{\log N} \neq \log M - \log N$. A quotient **of** logs is a change of base, not a difference.
- $(\log M)^k \neq k \log M$. The power law needs the exponent **inside** the log, on M.
- $\log(2x)$ is not $2\log x$. It is $\log 2 + \log x$.

Quick test: plug in numbers. $\log_{10}(10+10) = \log_{10} 20 \approx 1.301$, but $\log_{10} 10 + \log_{10} 10 = 2$.



Slide rules and log tables

John Napier published logarithms in 1614, and Henry Briggs soon produced base-10 tables. For three and a half centuries, scientists multiplied by looking up two logs, adding them, and looking up the antilog. A **slide rule** builds the same idea into hardware. Its scales are marked at distances proportional to $\log x$, so sliding one scale along another adds lengths, which multiplies numbers. The engineers on the Apollo programme carried slide rules. Laplace said logarithms, "by shortening the labours, doubled the life of the astronomer."

12.5. Solving exponential and logarithmic equations

12.5.1. The problem

Now we can answer the questions we set aside earlier. When will \$1,000 at 6% become \$2,000? How old is a bone with 30% of its carbon-14 left? In each the unknown is in the exponent, and the way to get it down is to **take a logarithm of both sides**.

12.5.2. Why it works

Taking logs is allowed because $\log$ is a function: equal inputs give equal outputs, so if $A = B$ (both positive) then $\log A = \log B$. Then the power law moves the unknown exponent down to the main line, where ordinary algebra (Chapter 5) can reach it:

$$b^x = c \implies \ln(b^x) = \ln c \implies x \ln b = \ln c \implies x = \frac{\ln c}{\ln b}.$$

The arrow $\implies$ means "implies".

12.5.3. The method

Example 70. Procedure — Solving an exponential equation

Input: an equation with the unknown in one or more exponents.

Steps:

1. **Isolate** the exponential term $b^{(\cdots)}$ on one side, using ordinary algebra (subtract constants, divide by coefficients).
2. **Same-base shortcut:** if both sides can be written as powers of the same base, set the exponents equal and skip to step 5.
3. **Take logs** of both sides. $\ln$ is the usual choice, or $\log_b$ when the base is b, since that cancels in one step.
4. **Bring down** the exponent with the power law.
5. **Solve** the resulting linear (or occasionally quadratic) equation for the unknown.
6. **Check** by substituting back. Also confirm the answer is sensible: a time that comes out negative, say, means something went wrong.

Output: exact solution(s) in terms of logs, and a decimal approximation.

12.5.4. Worked example 1: a pure exponential

Solve $2^x = 50$.

$$\begin{aligned} 2^x &= 50 && \\ \ln(2^x) &= \ln 50 && \text{(take ln of both sides)} \\ x \ln 2 &= \ln 50 && \text{(power law)} \\ x &= \frac{\ln 50}{\ln 2} && \text{(divide by } \ln 2\text{)} \\ &= \frac{3.912023}{0.693147} \approx 5.643856. \end{aligned}$$

Check: $2^5 = 32$ and $2^6 = 64$, so the answer has to be between 5 and 6. $x \approx 5.64$ passes. (Equivalently, $x = \log_2 50$.)

12.5.5. Worked example 2: isolate first

Solve $3 \cdot 5^{2x-1} = 400$.

$$
\begin{aligned}
3 \cdot 5^{2x-1} &= 400 \\
5^{2x-1} &= \frac{400}{3} \approx 133.3333 && \text{(isolate: divide by 3)} \\
(2x-1)\ln 5 &= \ln\left(\tfrac{400}{3}\right) && \text{(take ln, power law)} \\
2x - 1 &= \frac{\ln(400/3)}{\ln 5} = \frac{4.892852}{1.609438} \approx 3.040100 \\
2x &\approx 4.040100 && \text{(add 1)} \\
x &\approx 2.020050 && \text{(divide by 2).}
\end{aligned}
$$



Isolate before taking logs

A frequent wrong move is to take logs straight away and write $\ln(3 \cdot 5^{2x-1}) = (2x-1)\ln 15$. The 3 is **not** part of the base. Either isolate first (as above) or use the product law properly: $\ln 3 + (2x-1)\ln 5 = \ln 400$. Both routes give the same answer. And if the equation is $5^x + 3 = 20$, subtract the 3 **before** logging, because $\ln(5^x + 3)$ can't be simplified.

12.5.6. Worked example 3: carbon dating

A bone has 30% of the carbon-14 that living tissue has. How old is it? (Half-life 5,730 years.)

$$
\begin{aligned}
N_0\left(\tfrac{1}{2}\right)^{t/5730} &= 0.30\,N_0 && \text{(model = observation)} \\
\left(\tfrac{1}{2}\right)^{t/5730} &= 0.30 && \text{(divide by } N_0\text{)} \\
\frac{t}{5730}\ln\tfrac{1}{2} &= \ln 0.30 && \text{(take ln, power law)} \\
t &= 5730 \cdot \frac{\ln 0.30}{\ln 0.5} && \text{(rearrange)} \\
&= 5730 \cdot \frac{-1.203973}{-0.693147} \\
&= 5730 \times 1.736966 \approx 9{,}953 \text{ years.}
\end{aligned}
$$

The two negative logs cancel to give a positive age. Sanity check: 30% is between 50% (one half-life) and 25% (two half-lives), and 9,953 years is between 5,730 and 11,460.

12.5.7. Worked example 4: doubling time and the rule of 72

How long does money take to double at 6% compounded annually?

$$
\begin{aligned}
(1.06)^t &= 2 \\
t\ln 1.06 &= \ln 2 \\
t &= \frac{\ln 2}{\ln 1.06} = \frac{0.693147}{0.058269} \approx 11.90 \text{ years.}
\end{aligned}
$$

Bankers use the **rule of 72**: doubling time $\approx 72/(\text{rate in percent})$, here $72/6 = 12$. Why does it work? For small r, $\ln(1+r) \approx r$ (you'll see why in Volume 2, Chapter 17), so

$$
t = \frac{\ln 2}{\ln(1+r)} \approx \frac{0.693}{r} = \frac{69.3}{100r}.
$$

69.3 would be the more accurate constant for small rates. 72 is used because it divides evenly by 2, 3, 4, 6, 8, 9 and 12, and because it partly corrects for $\ln(1+r)$ being a little less than r at typical rates of 6–10%.

12.5.8. Worked example 5: a logarithmic equation

Solve $\log_2 x + \log_2(x-2) = 3$.

$$\begin{aligned}
\log_2\big(x(x-2)\big) &= 3 && \text{(product law, condense)}\\
x(x-2) &= 2^3 = 8 && \text{(definition: rewrite in exponential form)}\\
x^2 - 2x - 8 &= 0 && \text{(expand, rearrange)}\\
(x-4)(x+2) &= 0 && \text{(factor)}\\
x &= 4 \quad \text{or} \quad x = -2.
\end{aligned}$$

(Factoring quadratics is covered in Chapter 8.)

Check: $x = 4$ gives $\log_2 4 + \log_2 2 = 2 + 1 = 3$. Correct. $x = -2$ gives $\log_2(-2)$, which is undefined, so we reject it. It is an **extraneous solution**: condensing the logs widened the domain, because $x(x-2) > 0$ also allows $x < 0$.



Extraneous solution

An **extraneous solution** is a value produced by a valid-looking algebraic process that does not actually satisfy the original equation. Steps such as squaring both sides or combining logarithms can introduce them, because they can enlarge the set of allowed inputs. **Always substitute back into the original equation.**

12.5.9. Alternative approaches and choosing a method

- **Same-base matching.** $4^x = 8$ becomes $2^{2x} = 2^3$, so $x = 3/2$. You get an exact answer and need no calculator. Use it whenever both sides are obviously powers of one base.
- **Logarithms.** This is the general method for a single exponential term, or one exponential on each side ($3^x = 7^{x-1}$ becomes $x\ln 3 = (x-1)\ln 7$, which is linear in x).
- **Substitution.** $4^x - 3\cdot 2^x - 4 = 0$ is a quadratic in disguise. Let $u = 2^x$, so $4^x = u^2$, giving $u^2 - 3u - 4 = 0$, $(u-4)(u+1) = 0$. $u = 2^x$ must be positive, so $u = 4$ and $x = 2$.
- **Numerical or graphical methods.** Equations that mix an exponential with a polynomial, such as $2^x = x^{10}$ or $e^x = 3x$, **cannot** be solved with logs and algebra alone. Plot both sides, or use bisection or Newton's method (Volume 2, Chapter 32). Log-transforming can still help: $2^x = x^{10}$ is equivalent to $x\ln 2 = 10\ln x$, a nicer function to bisect.

The decision rule: **same base → match exponents; one exponential term → isolate and log; exponentials of related bases → substitute; exponential mixed with polynomial → numerical.**

12.5.10. Implementation

```python
import math

def solve_exponential(a, b, c):
    """Solve a * b**x = c for x. Requires a, c of the same sign, b > 0, b != 1."""
    if b <= 0 or b == 1:
        raise ValueError("base must be positive and not 1")
    ratio = c / a                    # step 1: isolate b**x
    if ratio <= 0:
        raise ValueError("no real solution: b**x is always positive")
    return math.log(ratio) / math.log(b)   # steps 3-5

x = solve_exponential(3, 5, 400)      # 3 * 5**x = 400
print(round(x, 6), 3 * 5**x)
print(round(solve_exponential(1, 1.06, 2), 4))   # doubling time at 6%
```

```
3.0401 400.0000000000001
11.8957
```

(The first line solves for the whole exponent $2x - 1 = 3.0401$. The final digit in `400.0000000000001` is floating-point rounding.)

12.5.11. Where you'd use this

Every "how long until …?" question about an exponential process is solved this way:

- **Debt.** An unpaid credit-card balance at 2% a month (about 24% APR) doubles when $1.02^t = 2$, i.e. after $t = \ln 2 / \ln 1.02 \approx 35$ months: under three years.
- **Medicine.** With caffeine's 5-hour half-life, when does a 200 mg dose fall below 50 mg? $200(\tfrac{1}{2})^{t/5} = 50$ gives $(\tfrac{1}{2})^{t/5} = \tfrac{1}{4}$, so $t/5 = 2$ and $t = 10$ hours. Pharmacologists use the same equation to decide when a drug has effectively cleared (about 5 half-lives, when under 4% remains).
- **Technology forecasts.** If computing power doubles every 2 years, how long until it is 1,000 times greater? $2^{t/2} = 1000$ gives $t = 2 \log_2 1000 \approx 19.9$ years: roughly 20 years for three orders of magnitude.

US curriculum

Exponential growth and decay models appear first in **Algebra I** (grades 8–9), usually with tables and graphs only. Logarithms, the log laws, e, and solving exponential equations are core **Algebra II** content (grades 10–11), revisited in **Precalculus**. AP Calculus assumes fluency with e^x and $\ln x$.

12.6. Logarithmic scales

12.6.1. The problem

The quietest sound a human can hear and the sound of a jet engine at 30 m differ in intensity by a factor of about 10^{14}. Earthquakes span a similar range of energies. Hydrogen-ion concentrations in everyday liquids run from about 1 down to 10^{-14} moles per litre. Put any of these on an ordinary linear axis and everything except the biggest values gets squashed into an invisible sliver near zero.

12.6.2. Intuition

A **logarithmic scale** places each value at a position proportional to its **logarithm**. Equal distances then represent equal **ratios**, not equal differences. On a base-10 log scale, 1 → 10 → 100 → 1,000 are evenly spaced, each one "decade" apart. Our senses work roughly like this: a sound ten times as intense doesn't seem ten times as loud.

12.6.3. Three famous scales

Decibels (sound, signals). The sound intensity level is *Try it:* Decibels and power ratios .

$$L = 10 \log_{10}\left(\frac{I}{I_0}\right) \text{ dB},$$

where $I_0 = 10^{-12}\ \text{W/m}^2$ is the threshold of hearing. Multiplying intensity by 10 adds 10 dB, and doubling it adds $10 \log_{10} 2 \approx 3.01$ dB. Engineers use dB for any power ratio: amplifier gain, Wi-Fi signal strength (dBm), the signal-to-noise ratio of an ADC.

Worked example. A sound is 85 dB. How many times more intense than a 60 dB sound is it? *Try it:* 25 dB as a ratio .

$$\begin{aligned} 85 - 60 &= 10 \log_{10}\frac{I_1}{I_0} - 10 \log_{10}\frac{I_2}{I_0} \\ 25 &= 10 \log_{10}\frac{I_1}{I_2} \qquad \text{(quotient law: the } I_0 \text{ cancels)} \\ \log_{10}\frac{I_1}{I_2} &= 2.5 \\ \frac{I_1}{I_2} &= 10^{2.5} \approx 316. \end{aligned}$$

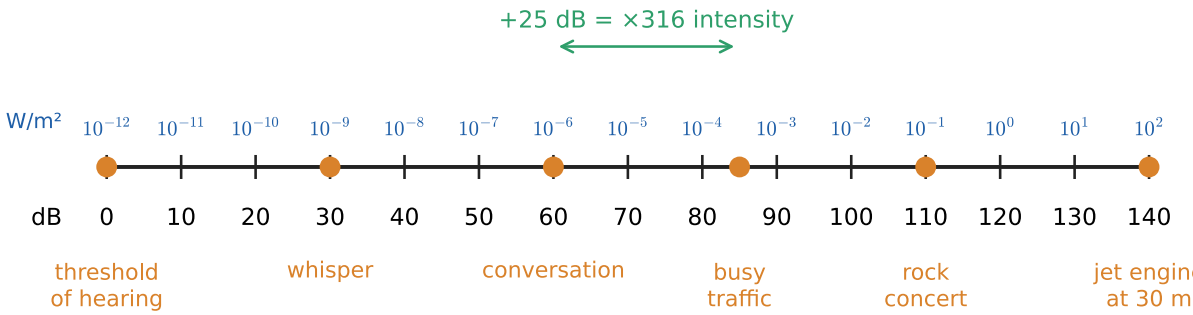


*Figure 96. The decibel scale. Each 10 dB step is the same **factor** of 10 in intensity (blue), so a range of 10^{14} fits on one readable line.*

Earthquake magnitude. The Richter scale, and the moment-magnitude scale that replaced it, is logarithmic. Each whole number up means about 10 times the ground-motion amplitude and about $10^{1.5} \approx 31.6$ times the energy released. A magnitude 7 quake releases about $31.6^2 \approx 1000$ times the energy of a magnitude 5.

pH (chemistry).

$$\mathrm{pH} = -\log_{10}[\mathrm{H}^+],$$

where $[\mathrm{H}^+]$ is the hydrogen-ion concentration in moles per litre. The minus sign makes everyday values positive. Pure water has a hydrogen-ion concentration of 10^{-7}, so pH 7. Lemon juice at pH 2 has $10^5 = 100{,}000$ times the hydrogen-ion concentration of water.



Logarithmic scale

A **logarithmic scale** positions a value v at distance proportional to $\log v$. Equal steps along it are equal **multiplications**. A **decade** is a factor of 10, and an **octave** (in music, and in frequency analysis) is a factor of 2.

12.6.4. Log plots: semi-log and log-log

Graphs can use a log scale on one axis or both.

- **Semi-log plot** (log scale on the y-axis only). An exponential $y = ab^x$ becomes a **straight line**: $\log y = \log a + x \log b$, which is linear in x with slope $\log b$. Use it to check whether data is exponential and to read off growth rates. Epidemic curves and Moore's-law charts are drawn this way.
- **Log-log plot** (both axes logarithmic). A power law $y = ax^k$ becomes a straight line: $\log y = \log a + k \log x$, with slope k. Use it to find exponents in power laws, for example when checking the empirical complexity of an algorithm.

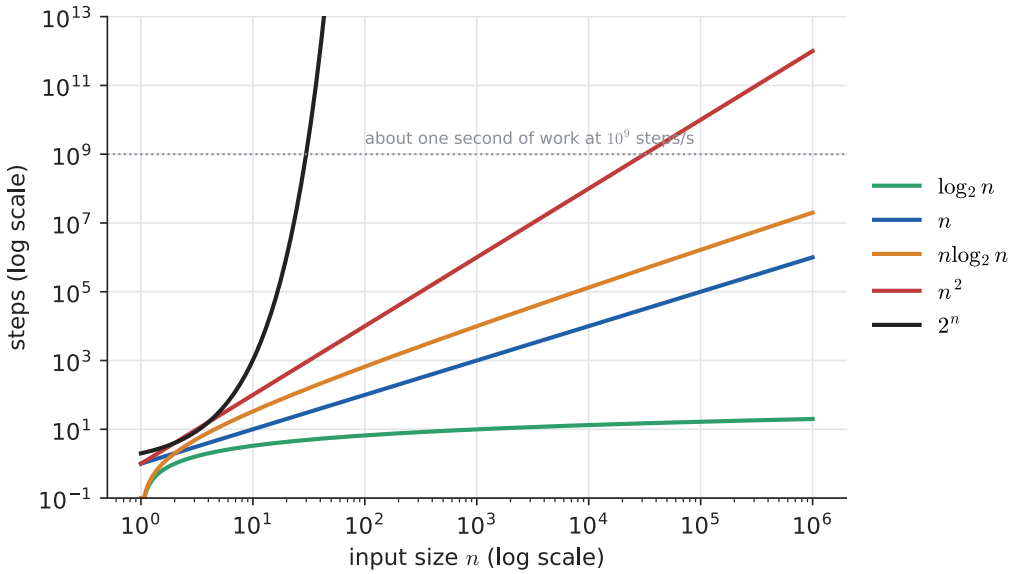


Figure 97. The growth-rate curves on log–log axes. Polynomials n, n^2 are straight lines whose slope is their degree; 2^n bends upward and off the chart.

A log-log plot is a practical tool for performance work. Time your function at $n = 1000, 2000, 4000, \ldots$, plot time against n on log-log axes, and measure the slope. A slope near 1 means linear, near 2 means quadratic. A slope that keeps increasing means something worse than polynomial is going on.

Reading log axes

- Zero and negative values can't appear on a log axis. Plotting libraries quietly drop them or warn.
- On a log scale, a small-looking gap can be an enormous difference. The distance from 10^6 to 10^9 looks the same as from 1 to 1,000.
- "Decibels" are always a **ratio** to some reference. "50 dB" means nothing until you know the reference (I_0, 1 mW, full scale, …).

Benford's law

In many real-world datasets (populations, river lengths, invoice amounts) the leading digit is 1 about 30.1% of the time and 9 only about 4.6% of the time. The probability of leading digit d is $\log_{10}(1 + 1/d)$: $\log_{10} 2 \approx 0.301$ for $d = 1$. This happens when data is spread evenly on a **log** scale. Forensic accountants use departures from Benford's law to flag fabricated figures.

12.7. Exponential models and fitting them to data

12.7.1. The problem

You have data, say case counts of a spreading infection, users of a new app, or the runtime of a brute-force solver as the input grows. It curves upward. Is it exponential? If so, what are a and b, what is the doubling time, and what does the model predict?

12.7.2. Intuition: straighten the curve

You know how to fit a straight line (Chapter 5). Logarithms turn an exponential into a straight line, so take the log of the data, fit a line, and convert back.

If $y = ae^{kt}$, then

$$\ln y = \ln a + kt.$$

This is a line in the variables t and $\ln y$, with slope k and intercept $\ln a$. The base-e form ae^{kt} is the one scientists prefer. It is the same family as ab^t, with $b = e^k$ (so $k = \ln b$). k is called the **continuous growth rate**.

12.7.3. Worked example: fitting by hand from two points

A culture has 500 cells at hour 2 and 4,000 cells at hour 5. Find $y = ae^{kt}$. *Try it:* Fit y = a·e^(kt) .

$$\begin{aligned}
ae^{5k} &= 4000, \quad ae^{2k} = 500 \\
\frac{ae^{5k}}{ae^{2k}} &= \frac{4000}{500} && \text{(divide to eliminate } a\text{)} \\
e^{3k} &= 8 && \text{(exponent law; } a \text{ cancels)} \\
3k &= \ln 8 && \text{(take ln)} \\
k &= \tfrac{\ln 8}{3} = \tfrac{3\ln 2}{3} = \ln 2 \approx 0.693147 \text{ per hour} \\
a &= \frac{500}{e^{2k}} = \frac{500}{e^{2\ln 2}} = \frac{500}{4} = 125 && \text{(since } e^{2\ln 2} = (e^{\ln 2})^2 = 4\text{).}
\end{aligned}$$

So $y = 125\,e^{0.6931\,t} = 125 \cdot 2^t$. The culture doubles every hour, and there were 125 cells at $t = 0$.

12.7.4. Worked example: fitting many points

With real, noisy data, use all the points. Fit a least-squares line to $(t, \ln y)$. (Least squares is covered properly in Volume 3, Chapter 31 . For now, think of it as "the line that best goes through the cloud of points".) *Try it:* Fit the case counts .

```python
import numpy as np

days  = np.arange(0, 11)
cases = np.array([12, 17, 26, 35, 54, 79, 110, 164, 237, 345, 502])

k, ln_a = np.polyfit(days, np.log(cases), 1)   # fit ln y = k t + ln a
a = np.exp(ln_a)
print(f"k = {k:.4f} per day, a = {a:.2f}")
print(f"daily growth factor = {np.exp(k):.4f}")
print(f"doubling time = {np.log(2) / k:.2f} days")
print(f"model at day 14: {a * np.exp(k * 14):.0f}")
```

Try it: Run this in your browser .

```
k = 0.3740 per day, a = 11.91
daily growth factor = 1.4535
doubling time = 1.85 days
model at day 14: 2238
```

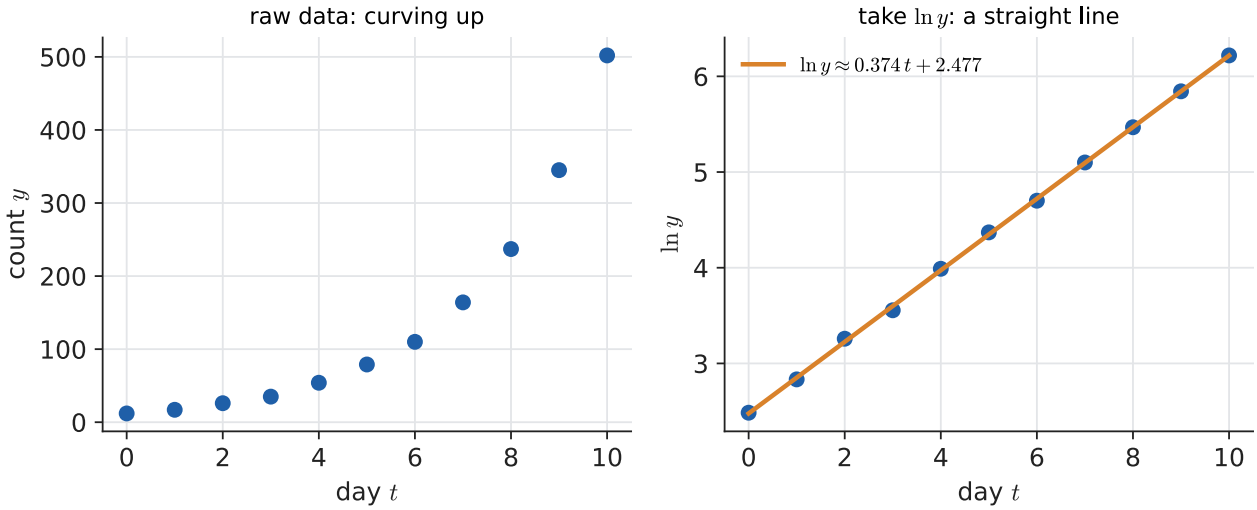


Figure 98. Left: counts that curve upward. Right: the same data after taking $\ln y$ lies on a line, so the model is exponential.

12.7.5. Choosing a model

How can you tell whether to use a linear, exponential or power-law model? Transform the data and see which version comes out straight:

Plot	Straight line means	Model
y vs x	constant differences	linear $y = mx + c$
$\ln y$ vs x (semi-log)	constant ratios	exponential $y = ae^{kx}$
$\ln y$ vs $\ln x$ (log-log)	constant elasticity	power law $y = ax^{p}$

12.7.6. Failure modes



Exponential growth always ends

Nothing physical grows exponentially for ever. Infections run out of susceptible people, bacteria run out of food, and apps run out of potential users. Real growth usually follows an S-shaped **logistic** curve. It looks exponential early on, then bends over and levels off at a **carrying capacity**. Extrapolating an exponential fit far beyond the data is one of the commonest forecasting errors there is. The rumour from the start of this chapter can't reach more than 8 billion people.

A technical caveat: fitting a line to $\ln y$ gives every point equal weight on the **log** scale. That is reasonable when the noise is proportional to the value, but it is not the same as least squares on y itself.



Exponential decay all around you

Newton's law of cooling (a hot coffee's excess temperature over the room decays exponentially), drug elimination from the bloodstream, a capacitor discharging through a resistor, light dimming as it passes through water, and the "exponential moving average" behind smoothed metrics on a dashboard all follow $y = ae^{-kt}$. They share a mechanism: **the rate of change is proportional to the current amount**. In Volume 2, Chapter 17 that becomes the differential equation $y' = ky$, and e^{kt} is its solution.



Two-point exponential fit

Drag points on a scatter plot and switch the y-axis between linear and log scale. Watch exponential data straighten under the log. Fit a line on the log view and see the fitted a and k (and doubling time) update live. Then add a logistic "saturation" and watch the fit's predictions go badly wrong beyond the data.

Try it: Exponential fit to data

12.8. Why exponentials beat polynomials

12.8.1. The problem

x^{10} is a huge polynomial. At $x = 10$ it is ten billion, while 2^{10} is only 1,024. Is x^{10} bigger for good? An algorithm that takes n^{10} steps would be useless in practice, but theory still calls it "efficient" (polynomial), while 2^n is "intractable". Why does exponential rank as worse?

12.8.2. Intuition: ratios

Compare how each function changes when x goes up by 1:

$$\frac{2^{x+1}}{2^x} = 2 \quad \textbf{always,} \qquad \frac{(x+1)^{10}}{x^{10}} = \left(1 + \frac{1}{x}\right)^{10} \to 1 \text{ as } x \to \infty.$$

The exponential multiplies itself by 2 at every step for ever. The polynomial's step-to-step growth factor **shrinks towards 1**. Eventually it is below 2 and stays there. Once the polynomial is growing by less than 2× per step and the exponential by exactly 2×, the exponential closes any gap and pulls ahead.

12.8.3. Making it precise with logs

Take logs of both to compare them:

$$\ln(2^x) = x \ln 2 \qquad \text{versus} \qquad \ln(x^{10}) = 10 \ln x.$$

The left side is a straight line. The right side is a log curve, which grows more slowly than any line. A line with any positive slope eventually overtakes $10 \ln x$ and never falls back. For 2^x vs x^{10}, solving $x \ln 2 = 10 \ln x$ numerically gives crossover points at $x \approx 1.078$ and $x \approx 58.77$. Beyond about 58.8, 2^x wins for ever.

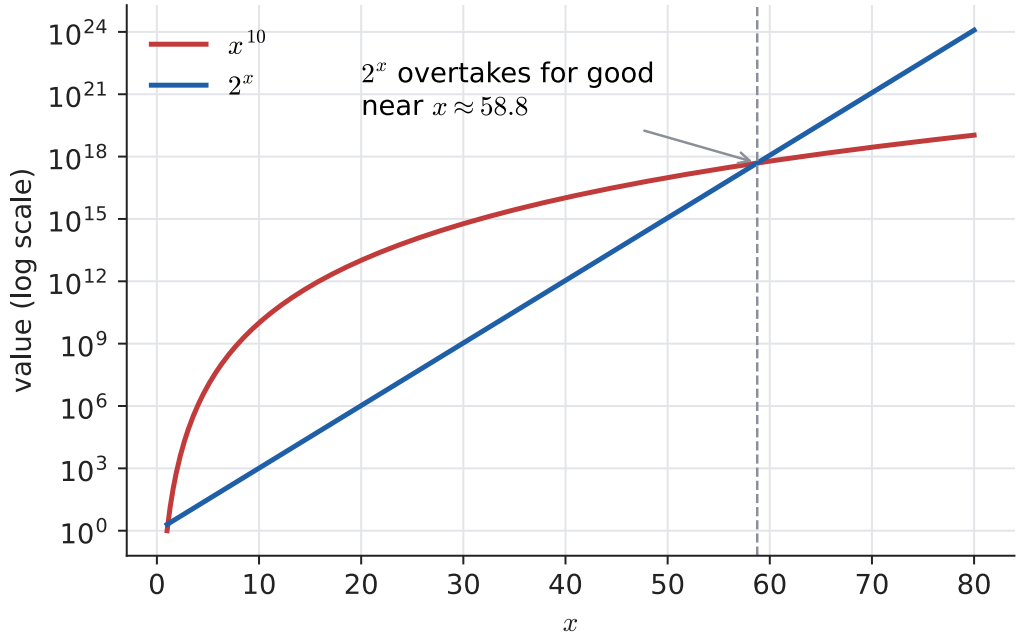


Figure 99. x^{10} leads for a long time, but 2^x overtakes near $x \approx 58.8$ and then grows much faster (log scale).

The general statements, which Volume 2, Chapter 16 makes rigorous:

- For any base $b > 1$ and any power p, **however large**: b^x eventually exceeds x^p.
- For any power $p > 0$, **however small**: x^p eventually exceeds $\log x$. Even $x^{0.01}$ beats $\ln x$ in the end, though only for astronomically large x.

That gives the **growth hierarchy**:

$$\log n \ll n^p \ (p > 0) \ll b^n \ (b > 1) \ll n! \ll n^n,$$

where $\ll$ means "is eventually negligible compared with".

12.9. Logarithms in computer science

This section collects the places where logarithms do real work in computing. The single most useful idea is this:

$\log_2 n$ is the number of times you can halve n before reaching 1.

12.9.1. Binary search: counting halvings

Searching a sorted array of n items, each comparison throws away half of what remains. After k comparisons, about $n/2^k$ items are left. We stop when one item is left: *Try it:* Binary search on a billion .

$$\frac{n}{2^k} = 1 \iff 2^k = n \iff k = \log_2 n.$$

So binary search needs about $\lceil \log_2 n \rceil$ comparisons. $\lceil \cdot \rceil$ is the **ceiling**, which rounds up to the next integer. For a billion items that is

$$\lceil \log_2 10^9 \rceil = \lceil 9 \log_2 10 \rceil = \lceil 29.897 \rceil = 30$$

comparisons, where linear search needs up to a billion.

Example 71. Procedure — Estimating $\log_2 n$ in your head

Input: a positive number n.

Steps:

1. Write n in powers of 1,000: $n \approx m \times 1000^j$.
2. Each factor of 1,000 contributes about 10 (because $2^{10} = 1024 \approx 1000$), so add $10j$.
3. Add $\log_2 m$ for the leftover m ($\log_2 2 = 1$, $\log_2 4 = 2$, $\log_2 8 = 3$, ..., $\log_2 500 \approx 9$).

Output: $\log_2 n \approx 10j + \log_2 m$. Example: 64 million $= 64 \times 1000^2$, so $\approx 20 + 6 = 26$. (The exact value is 25.93.)

Try it: Estimate $\log_2$ in your head .

12.9.2. Tree height

A **complete binary tree** of height h (h levels of edges below the root) holds up to *Try it:* Tree height and B-trees .

$$1 + 2 + 4 + \cdots + 2^h = 2^{h+1} - 1$$

nodes. (That sum is a geometric series, Chapter 14.) So a balanced tree holding n keys needs height only about $\log_2 n$. That is why a balanced binary search tree, a heap or a red–black tree gives $O(\log n)$ lookups and inserts. A B-tree with branching factor B has height about $\log_B n$. With

$B = 500$, a billion keys fit in a tree of height $\log_{500} 10^9 = 9 / \log_{10} 500 \approx 3.3$, so in practice four levels, four disk reads. *Try it:* Merge sort's $\log_2 n$ levels .

Merge sort splits the list in half repeatedly, giving $\log_2 n$ levels of splitting with n work per level, for a total of $O(n \log n)$ (Volume 3, Chapter 24 counts this more carefully).

12.9.3. Bits needed to represent n

With k bits you can write 2^k different values, 0 to $2^k - 1$. To store the integers 0 to n you need the smallest k with $2^k > n$:

$$k = \lfloor \log_2 n \rfloor + 1 \quad (n \geq 1),$$

where $\lfloor \cdot \rfloor$ is the **floor**, which rounds down. For example, 1,000,000 needs $\lfloor 19.93 \rfloor + 1 = 20$ bits. Likewise the number of decimal digits in n is $\lfloor \log_{10} n \rfloor + 1$.

PYTHON

```python
import math

for n in [1, 255, 256, 1_000_000, 10**9]:
    print(n, math.floor(math.log2(n)) + 1, n.bit_length())
```

```
1 1 1
255 8 8
256 9 9
1000000 20 20
1000000000 30 30
```



Floating-point logs at exact powers

`math.floor(math.log2(n)) + 1` is fine for moderate n. For very large integers near a power of 2, though, the floating-point log can land just below an integer and give a result that's off by one. `math.log10(10**15 - 1)` returns `15.0` even though the true value is slightly less than 15. Use exact integer operations where they exist (`int.bit_length()`, `len(str(n))`) for bit and digit counts.

12.9.4. Comparing growth rates

Here is how many basic steps algorithms of different complexity need, and how long they'd take at 10^9 steps per second:

n	$\log_2 n$	n	$n \log_2 n$	n^2	2^n
10	3.3	10	33	100	1,024
100	6.6	100	664	10,000	1.3×10^{30}
1,000	10.0	1,000	9,966	10^6	1.1×10^{301}
10^6	19.9	10^6	2.0×10^7	10^{12} (17 min)	—
10^9	29.9	10^9 (1 s)	3.0×10^{10} (30 s)	10^{18} (32 years)	—

2^{100} steps at a billion per second would take about 4×10^{13} years, thousands of times the age of the universe. That is why "exponential time" means "infeasible" for anything but small inputs.

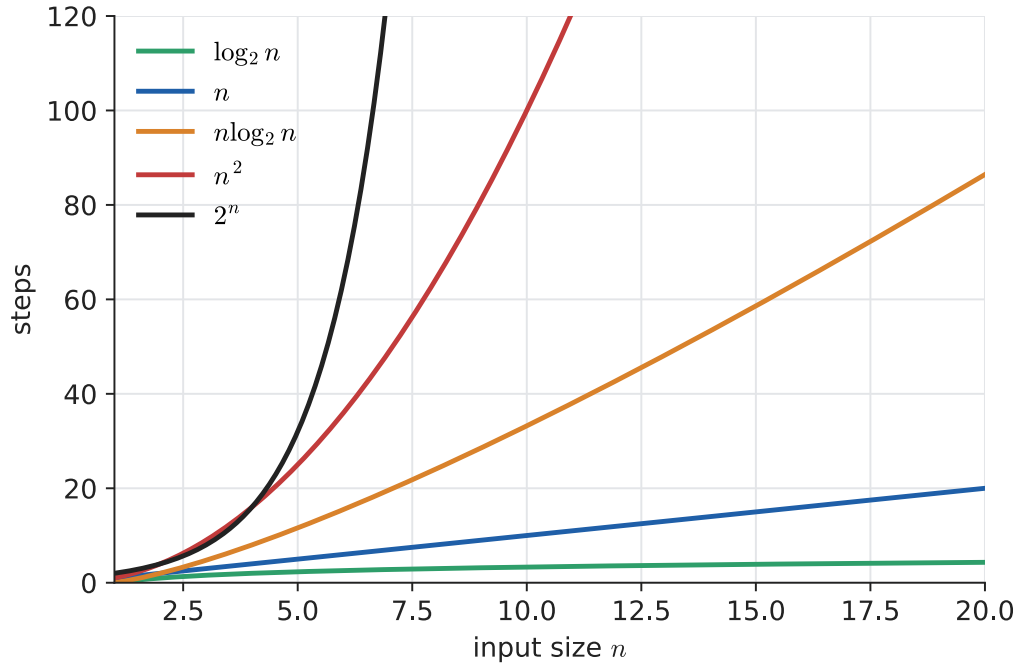


Figure 100. The same functions on ordinary axes. 2^n and n^2 leave the others behind even for small n.

Computer-science connection: why the base of the log doesn't matter in Big-O

You will see $O(\log n)$ written without a base. By change of base, $\log_2 n = \log_{10} n / \log_{10} 2 \approx 3.32 \log_{10} n$, and Big-O ignores constant factors, so $O(\log_2 n) = O(\log_{10} n) = O(\ln n)$. The base does matter for **actual** counts. A binary tree and a 256-way trie over the same keys differ in height by a factor of 8.

12.9.5. Information: how many bits is a choice worth?

How many yes/no questions does it take to identify one of 8 equally likely possibilities? Each well-chosen question halves the candidates: $8 \to 4 \to 2 \to 1$. That is **3** questions, and $\mathbf{3 = \log_2 8}$. In general, picking one of N equally likely options conveys

$$\log_2 N \text{ bits of information.}$$

A fair coin flip is 1 bit. A roll of a fair die is $\log_2 6 \approx 2.585$ bits. A random 12-character password from 94 printable characters has $12 \log_2 94 \approx 78.7$ bits of entropy. The game **Twenty Questions** can distinguish $2^{20} \approx 1{,}048{,}576$ objects.

Claude Shannon generalised this to unequal probabilities. An outcome of probability p carries $-\log_2 p$ bits of **surprise**, and the average surprise of a source is its **entropy**. That is the foundation of compression (Huffman coding, arithmetic coding) and of the cross-entropy loss used to train classifiers (Volume 3, Chapter 29).

12.9.6. Products to sums: log probabilities

Multiply a thousand probabilities of 0.01 each, as a language model or a naive Bayes classifier might when scoring a long document:

$$0.01^{1000} = 10^{-2000}.$$

The smallest positive double is about 5×10^{-324}, so the product **underflows** to 0.0, and every candidate document scores the same zero. The fix is to work in logs: the product law turns the product into a **sum**,

$$\ln \prod_{i} p_i = \sum_{i} \ln p_i = 1000 \ln 0.01 \approx -4605.17,$$

which is a perfectly ordinary floating-point number. ($\prod$ is the product symbol, the multiplicative cousin of $\sum$. Read $\prod_i p_i$ as "multiply all the p_i together". $\sum_i \ln p_i$ means "add up all the $\ln p_i$".) Since $\ln$ is increasing, comparing log-probabilities gives the same ranking as comparing probabilities. Log-likelihood, log-loss and log-odds all exist for this reason.

```python
import math

probs = [0.01] * 1000
product = 1.0
for p in probs:
    product *= p
print(product)                          # underflow
print(sum(math.log(p) for p in probs))  # fine
```

```
0.0
-4605.17018598815
```

12.9.7. A teaser: log-sum-exp and softmax

A classifier produces scores $z_1, \dots, z_K$ ("logits") and turns them into probabilities with the **softmax** function:

$$\mathrm{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j} e^{z_j}}.$$

If any z_i is around 1,000, then e^{1000} overflows to infinity and the result is `nan`. The standard fix uses an exponent law. For any constant m,

$$\frac{e^{z_i}}{\sum_j e^{z_j}} = \frac{e^{-m} e^{z_i}}{e^{-m} \sum_j e^{z_j}} = \frac{e^{z_i - m}}{\sum_j e^{z_j - m}}.$$

Choose $m = \max_j z_j$. Every exponent $z_j - m$ is then at most 0, so nothing overflows and at least one term equals 1. The same trick computes $\log \sum_j e^{z_j} = m + \log \sum_j e^{z_j - m}$ safely. This is the **log-sum-exp** function, which appears throughout machine learning.

PYTHON

```python
import numpy as np

def softmax_naive(z):
    e = np.exp(z)
    return e / e.sum()

def softmax_stable(z):
    e = np.exp(z - np.max(z))   # shift so the largest exponent is 0
    return e / e.sum()

z = np.array([1000.0, 1001.0, 1002.0])
with np.errstate(over="ignore", invalid="ignore"):
    print(softmax_naive(z))
print(softmax_stable(z))
```

```
[nan nan nan]
[0.09003057 0.24472847 0.66524096]
```

Computer-science connection: logarithms are everywhere in a codebase

- **Algorithms:** binary search, balanced trees, heaps, merge sort and quicksort ($n \log n$), divide-and-conquer recurrences, skip lists (expected $\log n$ levels).
- **Systems:** exponential backoff (wait 2^k ms after the k-th failure), log-structured merge trees, histogram buckets on a log scale (HDR histograms, latency percentiles), log-scale dashboards.
- **Numbers:** floating point stores a number as $m \times 2^e$, and the exponent field is essentially $\lfloor \log_2 |x| \rfloor$. Decibels in audio code. The Weber–Fechner reason why volume sliders should be logarithmic.
- **ML and IR:** log-likelihood, cross-entropy loss, softmax/log-sum-exp, TF–IDF (the "IDF" is $\log(N/n_t)$), log-odds in logistic regression.
- **Security:** bits of entropy in keys and passwords. The cost of brute force is 2^{bits}.



Halving explorer

Pick n with a slider (1 to 10^{12}) and watch binary search on a bar of n items. The bar halves on each step and a counter shows comparisons. Beside it, show $\log_2 n$, the bits needed for n, and the height of a balanced tree holding n keys, so the reader sees all three are the same number.

Try it: Halving explorer



Growth-race plotter

Plot $\log n$, n, $n \log n$, n^k (with k on a slider) and b^n (with b on a slider). Toggle linear, semi-log and log-log axes. Find where 1.01^n finally overtakes n^3. (It happens near $n \approx 2{,}340$.)

Try it: Growth race, log-log axes · Growth-race plotter · 1.01^n against n^3

12.10. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Decibels and gain chains
- Statistics
- Language sandbox
- Big-O explorer and benchmark
- Graphing calculator

12.11. Exercises

Recognition

1. Which of these are exponential functions? (a) $y = 3x^2$ (b) $y = 3 \cdot 2^x$ (c) $y = (0.7)^x$ (d) $y = x^x$ (e) $y = 5$.
2. The table $x = 0, 1, 2, 3$ has $y = 6, 9, 13.5, 20.25$. Linear or exponential? Find the formula.
3. Write in exponential form: (a) $\log_3 81 = 4$ (b) $\ln x = 2$. Write in log form: (c) $10^{-2} = 0.01$ (d) $e^0 = 1$.
4. Evaluate without a calculator: (a) $\log_2 64$ (b) $\log_{10} 0.0001$ (c) $\log_9 3$ (d) $\ln e^7$ (e) $\log_{27} 9$.

Application

5. Expand $\log_{10}\left(\dfrac{100\,x^2}{\sqrt[3]{y}}\right)$.
6. Condense to one log: $\frac{1}{2}\ln x - 2\ln(x+1) + \ln 3$.
7. Solve: (a) $5^x = 200$ (b) $2 \cdot 3^{x+1} = 54$ (c) $e^{2x} = 7$ (d) $4^x = 32$.
8. \$5,000 is invested at 4% a year. Find the balance after 15 years if compounded (a) annually (b) continuously. How long does it take to double in case (a)?
9. A drug has a half-life of 6 hours. What fraction remains after 24 hours? After how many hours is only 10% left?
10. Solve $\log_3(x+6) + \log_3 x = 3$, and check for extraneous solutions.

Problem solving

11. A sound meter reads 72 dB in a café and 96 dB at a concert. How many times more intense is the concert?
12. A sorted table has 50 million rows. At most how many comparisons does binary search need? How many bits are needed to store a row index?
13. Two measurements of a population: 2,000 in 2010, 2,900 in 2020. Fit $P = ae^{k(t-2010)}$. Predict 2030 and give the doubling time.
14. Show that $\log_b x \cdot \log_x b = 1$ for valid b, x.
15. A website's traffic doubles every 4 months. Write the monthly growth factor exactly, and give the percentage monthly growth rate.
16. An algorithm takes n^3 steps and another takes 1.5^n steps. Use logs and a quick search (by hand or in code) to estimate the largest n for which the exponential one is still faster, beyond the trivial region near $n = 1$.

▼ **Answers**

1. (b) and (c) are exponential. (a) is a power function. (d) has the variable in the base **and** the exponent, so it isn't of the form ab^x. (e) is constant ($b = 1$ is excluded).
2. Ratios: $9/6 = 13.5/9 = 20.25/13.5 = 1.5$. Constant ratio, so exponential: $y = 6(1.5)^x$.
3. (a) $3^4 = 81$ (b) $e^2 = x$ (c) $\log_{10} 0.01 = -2$ (d) $\ln 1 = 0$.
4. (a) 6 ($2^6 = 64$) (b) -4 (c) $\frac{1}{2}$ ($9^{1/2} = 3$) (d) 7 (e) $27^y = 9 \Rightarrow 3^{3y} = 3^2 \Rightarrow y = \frac{2}{3}$.
5. $\log 100 + \log x^2 - \log y^{1/3} = 2 + 2\log x - \frac{1}{3}\log y$.
6. $\ln \sqrt{x} - \ln(x+1)^2 + \ln 3 = \ln\left(\dfrac{3\sqrt{x}}{(x+1)^2}\right)$.
7. (a) $x = \ln 200 / \ln 5 = 5.298317/1.609438 \approx 3.292$. (b) $3^{x+1} = 27 = 3^3 \Rightarrow x + 1 = 3 \Rightarrow x = 2$. (c) $2x = \ln 7 \Rightarrow x = 1.945910/2 \approx 0.973$. (d) $2^{2x} = 2^5 \Rightarrow x = 5/2$.
8. (a) $5000(1.04)^{15} = 5000 \times 1.800944 \approx \$9{,}004.72$. (b) $5000e^{0.6} = 5000 \times 1.822119 \approx \$9{,}110.59$. Doubling: $t = \ln 2 / \ln 1.04 = 0.693147/0.039221 \approx 17.67$ years (rule of 72: 18).
9. 24 hours is 4 half-lives, so $(1/2)^4 = 1/16 = 6.25\%$ remains. For 10%: $(1/2)^{t/6} = 0.1 \Rightarrow t = 6 \ln 0.1 / \ln 0.5 = 6 \times 3.321928 \approx 19.93$ hours.
10. $\log_3\big(x(x+6)\big) = 3 \Rightarrow x^2 + 6x = 27 \Rightarrow x^2 + 6x - 27 = 0 \Rightarrow (x+9)(x-3) = 0$. $x = 3$ checks: $\log_3 9 + \log_3 3 = 2 + 1 = 3$. $x = -9$ is extraneous (log of a negative).
11. $96 - 72 = 24 = 10\log_{10}(I_1/I_2)$, so $I_1/I_2 = 10^{2.4} \approx 251$.
12. $\log_2(5 \times 10^7) \approx 25.58$, so 26 comparisons. Indices 0 to 49,999,999 need $\lfloor \log_2 49{,}999{,}999 \rfloor + 1 = 25 + 1 = 26$ bits.
13. $a = 2000$. $e^{10k} = 1.45 \Rightarrow k = \ln 1.45/10 = 0.371564/10 \approx 0.037156$. 2030: $2000e^{20k} = 2000 \times 1.45^2 = 4{,}205$. Doubling time $\ln 2/k \approx 18.65$ years.
14. By change of base, $\log_b x = \dfrac{\ln x}{\ln b}$ and $\log_x b = \dfrac{\ln b}{\ln x}$. Their product is 1.
15. Doubling in 4 months means $b^4 = 2$, so $b = 2^{1/4} \approx 1.189207$: about 18.92% a month.
16. Compare $n \ln 1.5$ with $3 \ln n$. At $n = 26$: $26 \times 0.405465 = 10.54$ versus $3 \ln 26 = 9.77$, so the exponential is already bigger. At $n = 25$: $10.14 > 9.66$, also bigger. At $n = 23$: $9.33 < 9.41$, smaller. At $n = 24$: $9.73 > 9.53$, bigger. So $1.5^n < n^3$ up to $n = 23$ ($1.5^{23} \approx 11{,}223$ vs $23^3 = 12{,}167$), and the exponential algorithm is slower from $n = 24$ onward. The largest n for which the exponential algorithm is still faster is 23.

12.12. Mastery check

1. Without a calculator: $\log_2 \frac{1}{32}$, $\log_{100} 10$, $e^{\ln 5}$, $\log_5 5^{-3}$.
2. Derive the quotient law $\log_b(M/N) = \log_b M - \log_b N$ from an exponent law.
3. Solve $7 \cdot 2^{3x} = 91$. Give an exact answer and a decimal.
4. A culture triples every 5 hours. How long does it take to grow 1,000-fold?
5. Solve $\ln(x) + \ln(x-3) = \ln 10$.
6. How many comparisons does binary search need on $2^{20} + 1$ sorted items, in the worst case? How many bits are needed to write the number 2^{20}?
7. Explain in two sentences why multiplying many small probabilities in floating point is dangerous, and what to do instead.

Decision rule: All correct without hesitation → move on to Chapter 13 (or Chapter 14 if trigonometry is solid). Stuck on 1–3 → reread Section 12.3 and Section 12.4. Stuck on 4–5 → reread Section 12.5. Stuck on 6–7 → reread Section 12.9. Stuck on more → work the section exercises first, and check exponent laws in Chapter 3.

▼ **Mastery check answers**

1. -5; $\frac{1}{2}$ (since $100^{1/2} = 10$); 5; -3.
2. Let $M = b^m$, $N = b^n$. Then $M/N = b^m / b^n = b^{m-n}$, so $\log_b(M/N) = m - n = \log_b M - \log_b N$.
3. $2^{3x} = 13 \Rightarrow 3x \ln 2 = \ln 13 \Rightarrow x = \dfrac{\ln 13}{3 \ln 2} = \dfrac{2.564949}{2.079442} \approx 1.2335$. (Exact: $x = \frac{1}{3}\log_2 13$.)
4. $3^{t/5} = 1000 \Rightarrow \frac{t}{5}\ln 3 = \ln 1000 \Rightarrow t = \dfrac{5 \times 6.907755}{1.098612} \approx 31.44$ hours.
5. $\ln\big(x(x-3)\big) = \ln 10 \Rightarrow x^2 - 3x - 10 = 0 \Rightarrow (x-5)(x+2) = 0$. $x = 5$ (check: $\ln 5 + \ln 2 = \ln 10$). $x = -2$ is extraneous.
6. $\lceil \log_2(2^{20} + 1) \rceil = 21$ comparisons. (Each comparison halves the range, and 20 halvings leave up to 2 candidates when there are $2^{20} + 1$ items. The exact worst case for the usual implementation is $\lfloor \log_2 n \rfloor + 1 = 21$.) 2^{20} in binary is a 1 followed by 20 zeros: 21 bits.
7. The product of many numbers below 1 gets smaller than the smallest representable double and underflows to 0, so all candidates look identical and later divisions or logs fail. Instead, add logarithms: $\sum \ln p_i$ carries the same ordering information in a safe numeric range.

Chapter 13. Trigonometry

Trigonometry began as a way to measure what you can't reach: the height of a tree, the width of a river, the distance to a ship or a star. It works because, in a right triangle, the ratios of the sides depend only on the angles. Extend those ratios from triangles to the full circle and they become the sine and cosine functions, the basic description of anything that rotates, oscillates or repeats. That covers pendulums and springs, sound and radio, and every rotation in a game engine. This chapter takes that whole path, from a measuring tape to atan2 and a first look at Fourier analysis.

Where this chapter sits

You need: similar triangles and Pythagoras' theorem (Chapter 9), coordinates and distance in the plane (Chapter 10), functions, graphs, transformations and inverses (Chapter 11).

This chapter unlocks: complex numbers in polar form and Euler's formula $e^{i\theta} = \cos\theta + i\sin\theta$ (Chapter 15); derivatives and integrals of sin and cos (Volume 2, Chapter 17 , Volume 2, Chapter 19); vectors, dot products and the angle between vectors (Volume 2, Chapter 26); rotation matrices (Volume 2, Chapter 27); periodic signals and their numerical treatment (Volume 2, Chapter 32).

If you are in a hurry: read Section 13.1, Section 13.2 and Section 13.3 carefully, since everything else depends on them. Programmers should not skip Section 13.9.

13.1. Measuring what you can't reach: right-triangle trigonometry

13.1.1. The problem

You want the height of a tree. You can't climb it, but you can walk away from its base. You pace out 30 m along flat ground, then use a phone's inclinometer app to measure the angle from the ground up to the treetop: 35°. That's all you have, one distance and one angle. Is it enough?

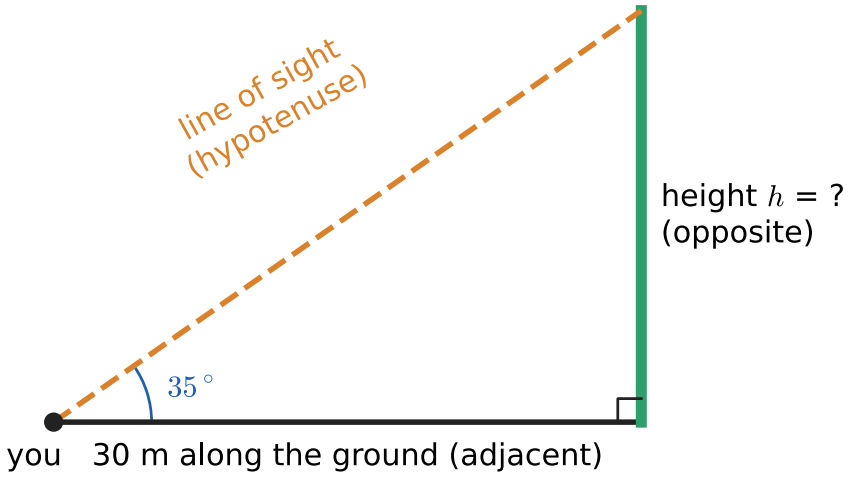


Figure 101. Thirty metres along the ground and a 35° sightline. The triangle is fixed, so the height is fixed too.

Yes, it is. Once you know the angle and the ground distance, only one right triangle fits, so the height is determined. You just need a way to compute it.

13.1.2. Intuition: shape is fixed by angles

Think back to **similar triangles** (Chapter 9). Two triangles with the same angles have the same shape, and one is a scaled copy of the other. Scaling multiplies every side by the same factor, so the **ratio** of any two sides doesn't change.

So for a right triangle with a 35° angle, the ratio **height/ground** is the same whether the triangle is drawn on a napkin or stretches across a valley. Measure it once on any 35° right triangle (the napkin will do) and you can use it for every 35° right triangle. People have tabulated these ratios for every angle, and now a calculator does it for you.

13.1.3. Vocabulary and notation



Opposite, adjacent, hypotenuse

In a right triangle, pick one of the two acute angles and call it θ (the Greek letter **theta**, the traditional name for an angle).

- The **hypotenuse** is the longest side, across from the right angle.
- The **opposite** side is the side across from θ.
- The **adjacent** side is the side next to θ that isn't the hypotenuse.

"Opposite" and "adjacent" depend on which angle you are looking from. Swap to the other acute angle and they swap too.

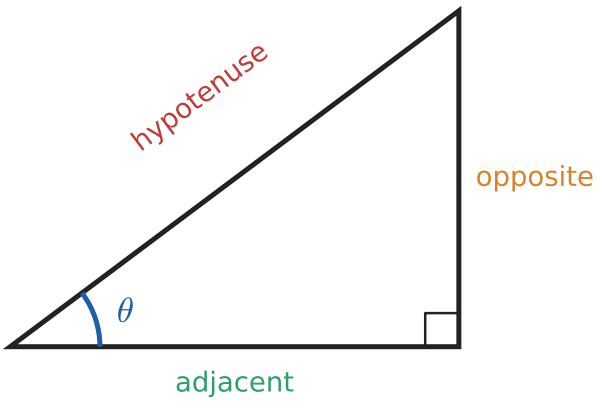


Figure 102. Labelling relative to the angle θ.

Sine, cosine, tangent

For an acute angle θ in a right triangle:

$$\sin\theta = \frac{\text{opposite}}{\text{hypotenuse}}, \qquad \cos\theta = \frac{\text{adjacent}}{\text{hypotenuse}}, \qquad \tan\theta = \frac{\text{opposite}}{\text{adjacent}}.$$

The mnemonic is **SOH-CAH-TOA**: **S**ine = **O**pposite/**H**ypotenuse, **C**osine = **A**djacent/**H**ypotenuse, **T**angent = **O**pposite/**A**djacent.

Dividing the first by the second gives $\tan\theta = \dfrac{\sin\theta}{\cos\theta}$.

Notation: $\sin\theta$ means "sine of θ". The brackets are optional for a single symbol but needed for anything longer: $\sin(2\theta+1)$. $\sin^2\theta$ means $(\sin\theta)^2$, **not** $\sin(\sin\theta)$.

Reciprocal ratios

Three more names you'll meet in older texts and in calculus:

$$\csc\theta = \frac{1}{\sin\theta}\ \text{(cosecant)}, \qquad \sec\theta = \frac{1}{\cos\theta}\ \text{(secant)}, \qquad \cot\theta = \frac{1}{\tan\theta}\ \text{(cotangent)}.$$

Note the pairing: *co*secant goes with sine, and secant goes with *co*sine. They add no new information, only convenient shorthand.

13.1.4. Worked example: the tree

The 30 m ground distance is **adjacent** to the 35° angle, and the height we want is **opposite** it. The ratio linking opposite and adjacent is tangent:

$$\begin{aligned} \tan 35^\circ &= \frac{h}{30} && \text{(TOA)} \\ h &= 30\tan 35^\circ && \text{(multiply both sides by 30)} \\ &= 30 \times 0.700208 && \text{(calculator, in degree mode)} \\ &\approx 21.0\ \text{m}. \end{aligned}$$

One refinement: your eye (or phone) isn't at ground level. If it is 1.5 m up, the triangle starts at that height, and the tree is about $21.0 + 1.5 = 22.5$ m tall. Real surveying is full of corrections like this.

13.1.5. The method

Example 72. Procedure — Solving a right triangle

Input: a right triangle with two known quantities besides the right angle, at least one of which is a side.

Steps:

1. Sketch the triangle. Mark the right angle, the known angle θ, and label the sides **opposite**, **adjacent** and **hypotenuse** relative to θ.
2. Identify the side you **know** and the side you **want**.
3. Choose the ratio that contains exactly those two: both legs → tan; a leg and the hypotenuse → sin (if the leg is opposite) or cos (if adjacent).
4. Write the equation and solve for the unknown (multiply if it's on top, divide if it's underneath).
5. If you know two sides and want an angle, form the ratio and apply the **inverse** function (below).
6. For the remaining pieces: the third side comes from Pythagoras ($a^2 + b^2 = c^2$), and the other acute angle is $90° - \theta$.
7. Check: the hypotenuse must be the longest side, and the larger angle must face the larger side.

Output: all three sides and all three angles.

13.1.6. Worked example: a ladder

A 6 m ladder leans against a wall at 70° to the ground. How high up the wall does it reach, and how far is its foot from the wall?

The ladder is the hypotenuse. The height on the wall is opposite the 70° angle, and the distance from the wall is adjacent to it.

$$\begin{aligned} \text{height} &= 6\sin 70° = 6 \times 0.939693 \approx 5.64 \text{ m} && \text{(SOH: opposite} = \text{hyp} \times \sin\text{)} \\ \text{foot} &= 6\cos 70° = 6 \times 0.342020 \approx 2.05 \text{ m} && \text{(CAH: adjacent} = \text{hyp} \times \cos\text{)} \\ \text{check: } & 5.638^2 + 2.052^2 = 31.79 + 4.21 = 36.00 = 6^2. && \checkmark \end{aligned}$$

13.1.7. Inverse trigonometric functions

Now the reverse problem. A wheelchair ramp rises 1 m over a horizontal run of 12 m. What angle does it make with the ground?

Here we know $\tan\theta = 1/12$ and want θ. The function that answers "which angle has this tangent?" is the **inverse tangent**, written $\tan^{-1}$ or **arctan**:

$$\theta = \tan^{-1}\left(\frac{1}{12}\right) = \tan^{-1}(0.083333) \approx 4.76^\circ.$$



Inverse trig functions

$\sin^{-1} x$ (also $\arcsin x$), $\cos^{-1} x$ ($\arccos x$) and $\tan^{-1} x$ ($\arctan x$) return **the angle** whose sine, cosine or tangent is x.

$$\theta = \sin^{-1} x \quad\Longleftrightarrow\quad \sin\theta = x \text{ and } \theta \text{ is in the principal range.}$$

The **principal ranges** are $-90^\circ \le \sin^{-1} x \le 90^\circ$, $0^\circ \le \cos^{-1} x \le 180^\circ$ and $-90^\circ < \tan^{-1} x < 90^\circ$. There are infinitely many angles with a given sine, so the inverse function has to pick one (see Section 13.3).

$\sin^{-1} x$ is **not** $1/\sin x$. The -1 here means "inverse function", as in f^{-1} (Chapter 11), not "reciprocal". This is a notorious notational clash, and it's why many people prefer writing arcsin.



Calculator mode and impossible ratios

- A calculator in **radian** mode returns $\tan 35 \approx 0.4738$ (the tangent of 35 **radians**), which is completely wrong for the tree problem. If a triangle answer looks absurd, check the mode first. Section Section 13.2 explains radians, and Section 13.9 covers the same bug in code.
- $\sin^{-1}(1.2)$ is undefined. No right triangle has an opposite side longer than its hypotenuse, so $|\sin\theta| \le 1$ and $|\cos\theta| \le 1$ always. If you get a sine above 1, there's an arithmetic error earlier on.
- SOH-CAH-TOA applies only to **right** triangles. For other triangles use the laws of sines and cosines (Section 13.6).

13.1.8. Special angles you can compute exactly

Three angles come up so often that their exact ratios are worth deriving once.

45°. Take a square with side 1 and cut it along a diagonal. Each half is a right isosceles triangle with legs 1, 1 and hypotenuse $\sqrt{1^2+1^2} = \sqrt{2}$:

$$\sin 45^\circ = \cos 45^\circ = \frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2} \approx 0.7071, \qquad \tan 45^\circ = 1.$$

30° and 60°. Take an equilateral triangle with side 2 (all angles 60°) and drop a perpendicular from the top vertex. It bisects the base, giving a right triangle with hypotenuse 2, short leg 1, angles 30° and 60°, and long leg $\sqrt{2^2-1^2} = \sqrt{3}$:

θ	$\sin\theta$	$\cos\theta$	$\tan\theta$
0°	0	1	0
30°	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{3}} = \frac{\sqrt{3}}{3}$
45°	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	1
60°	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$\sqrt{3}$
90°	1	0	undefined

A pattern to remember: down the sine column the values are $\frac{\sqrt{0}}{2}, \frac{\sqrt{1}}{2}, \frac{\sqrt{2}}{2}, \frac{\sqrt{3}}{2}, \frac{\sqrt{4}}{2}$, and cosine is the same list read upward. Cosine is the sine of the *co*mplementary angle: $\cos\theta = \sin(90° - \theta)$. That's where the "co" in cosine comes from.

13.1.9. Where you'd use this

- **Construction and DIY:** heights of buildings, roof pitch, the grade of a road. A 6% grade means $\tan\theta = 0.06$, so $\theta \approx 3.43°$. The US ADA maximum ramp slope of 1:12 is the 4.76° we just computed.
- **Shopping for a screen:** a "65-inch" 16:9 TV is measured on the diagonal. The diagonal makes angle $\theta = \tan^{-1}(9/16) \approx 29.36°$ with the bottom edge, so the width is $65\cos\theta \approx 56.7$ inches and the height $65\sin\theta \approx 31.9$ inches. That's what tells you whether it fits the alcove.
- **Mechanics:** a 5 kg box rests on a ramp at 30°. Its weight, $5 \times 9.81 = 49.05$ N, splits into $49.05\sin 30° \approx 24.5$ N pulling it down the slope and $49.05\cos 30° \approx 42.5$ N pressing into the surface. Friction has to supply the 24.5 N or the box slides. We develop components properly in Volume 2, Chapter 26 .
- **Computing:** a drone camera with an 84° field of view, hovering 120 m up and pointing straight down, sees a strip of ground $2 \times 120\tan 42° \approx 216$ m wide (half the angle, half the width, then double). Mapping software uses exactly this to plan how far apart to space the photos. Game engines use the same formula to decide what the camera can see.



> *Why "sine"?*
> The word "sine" comes from a chain of mistranslations. Indian astronomers (Aryabhata, around 500 CE) used **jyā-ardha**, "half-chord", shortened to **jyā**. Arabic scholars transliterated it as **jība**, written without vowels as **jb**. Later Latin translators read this as **jaib**, "fold" or "bosom", and translated it as the Latin **sinus**, which means the same thing. "Tangent" (Latin **tangens**, "touching") and "secant" (**secans**, "cutting") are named after lines that touch or cut a circle, as Section 13.3 shows.

US curriculum

Right-triangle trigonometry (SOH-CAH-TOA, special right triangles, inverse ratios) is part of high-school **Geometry** (usually grade 9 or 10). It is extended to the unit circle, radians and graphs in **Algebra II** or **Precalculus** (grades 10–12).

13.2. Radians: measuring angles by arc length

13.2.1. The problem

Why does a full turn have 360 degrees? The Babylonians liked base 60, and 360 is close to the number of days in a year and divides evenly by many numbers. Those are cultural reasons, not mathematical ones. If you had to pick a unit for angles that came from the geometry of the circle itself, what would you choose?

13.2.2. Intuition

An angle at the centre of a circle cuts off an arc. Larger angles cut off longer arcs, in proportion. On a circle of radius 1, the arc length could **be** the measure of the angle. On a circle of radius r everything scales by r, so the natural measure is

$$\theta = \frac{\text{arc length}}{\text{radius}} = \frac{s}{r}.$$

That's the **radian**. One radian is the angle whose arc is exactly one radius long. *Try it:* See one radian .

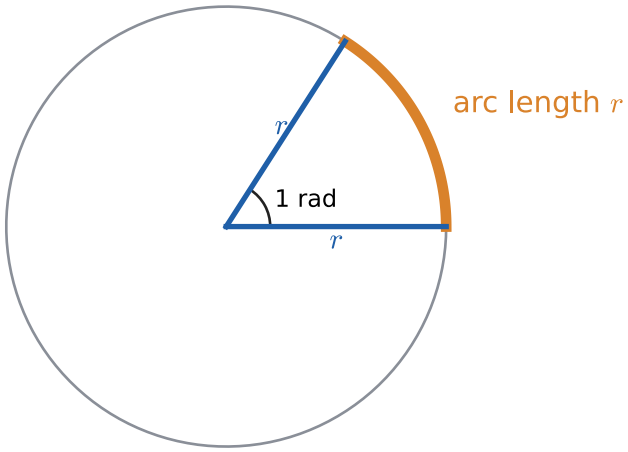


Figure 103. One radian: the arc (orange) has the same length as the radius.

13.2.3. Converting

A full circle has circumference $2\pi r$, so a full turn is $2\pi r / r = 2\pi$ radians. Therefore

$$360^\circ = 2\pi \text{ rad}, \qquad 180^\circ = \pi \text{ rad}, \qquad 1 \text{ rad} = \frac{180^\circ}{\pi} \approx 57.2958^\circ.$$

Radian

An angle of θ **radians** at the centre of a circle of radius r cuts off an arc of length $s = r\theta$. Radians are a ratio of two lengths, so they have no physical unit, and "rad" is often left out. **If no unit is written, assume radians.**

Converting: multiply degrees by $\dfrac{\pi}{180}$ to get radians, and multiply radians by $\dfrac{180}{\pi}$ to get degrees.

Common values: $30^\circ = \frac{\pi}{6}$, $45^\circ = \frac{\pi}{4}$, $60^\circ = \frac{\pi}{3}$, $90^\circ = \frac{\pi}{2}$, $180^\circ = \pi$, $270^\circ = \frac{3\pi}{2}$, $360^\circ = 2\pi$.

13.2.4. Worked examples

Convert 135° to radians: $135 \times \dfrac{\pi}{180} = \dfrac{135\pi}{180} = \dfrac{3\pi}{4} \approx 2.356$. *Try it:* Convert 135° interactively .

Convert $\frac{5\pi}{6}$ to degrees: $\dfrac{5\pi}{6} \times \dfrac{180}{\pi} = \dfrac{5 \times 180}{6} = 150^\circ$.

Arc length. A pizza with radius 18 cm is cut into 8 equal slices. How long is the crust edge of one slice? *Try it:* Arc length of the pizza slice .

$$\theta = \frac{2\pi}{8} = \frac{\pi}{4} \qquad \text{(one eighth of a full turn)}$$
$$s = r\theta = 18 \times \frac{\pi}{4} = 4.5\pi \approx 14.14 \text{ cm}.$$

Sector area. A sector with angle θ is the fraction $\theta/2\pi$ of the whole disc, so its area is $\dfrac{\theta}{2\pi} \cdot \pi r^2 = \frac{1}{2} r^2 \theta$. For the pizza slice: $\frac{1}{2} \times 18^2 \times \frac{\pi}{4} = 40.5\pi \approx 127.2 \text{ cm}^2$.

The nautical mile. The Earth's radius is about 6,371 km. One minute of arc ($1/60$ of a degree) along a great circle is

$$s = r\theta = 6371 \times \left(\frac{1}{60} \times \frac{\pi}{180}\right) \approx 1.853 \text{ km},$$

which is (to within rounding) the **nautical mile**, 1.852 km. Sea and air navigation still use it for this reason. One degree of latitude is about 111 km.

13.2.5. Why radians are natural

With degrees, $s = r\theta$ would need a conversion factor: $s = \frac{\pi}{180} r\theta$. Radians remove it. The real payoff comes in calculus. Look at small angles:

θ (radians)	$\sin\theta$	$\sin\theta/\theta$
0.5	0.479426	0.958851
0.1	0.099833	0.998334
0.01	0.0099998	0.999983

Try it: Watch sin θ/θ approach 1 .

For small θ **in radians**, $\sin\theta \approx \theta$. In Volume 2, Chapter 16 this becomes $\lim_{\theta\to 0}\frac{\sin\theta}{\theta} = 1$, and in Volume 2, Chapter 17 it gives the clean rule "the derivative of $\sin x$ is $\cos x$". In degrees the rule would read "the derivative of $\sin x$ is $\frac{\pi}{180}\cos x$", and that factor would follow you through every formula in physics. Radians are the unit in which the circle's own geometry needs no conversion constant.

The **small-angle approximation** $\sin\theta \approx \theta$ is also what makes the simple pendulum formula work (Section 13.8).

13.2.6. Where you'd use this

Radians are the natural unit whenever something **turns** and you care how far a point on it **travels**, because the link is simply $s = r\theta$, or per second, $v = r\omega$.

- **Cycling:** a road bike's wheel has radius about 0.35 m. At 25 km/h (6.94 m/s) the wheel turns at $\omega = v/r = 6.94/0.35 \approx 19.8$ rad/s, which is $19.8 \times 60/2\pi \approx 189$ revolutions per minute. A bike computer does this calculation in reverse: it counts wheel revolutions and multiplies by $2\pi r$.
- **Engineering:** a hard disk spinning at 7,200 rpm turns at $7200 \times 2\pi/60 \approx 754$ rad/s. A track 4.5 cm from the centre passes under the head at $0.045 \times 754 \approx 33.9$ m/s, more than 120 km/h. The outer tracks move faster than the inner ones, which is why they read faster.
- **Robotics:** a robot's wheel encoder gives 1,024 ticks per revolution and the wheel has radius 5 cm. Each tick is $2\pi/1024$ rad, so the robot moves $0.05 \times 2\pi/1024 \approx 0.307$ mm per tick. Counting ticks is how a robot vacuum knows how far it has gone (**odometry**).



Units, again

Every mainstream programming language's `sin` , `cos` , `tan` take **radians**, and `asin` , `acos` , `atan` , `atan2` **return** radians. Spreadsheet `SIN` uses radians too. Game engines and UI toolkits, on the other hand, often expose angles in **degrees** (Unity's `transform.eulerAngles` , CSS `rotate(45deg)` , SVG `rotate(45)`). Mixing the two is one of the commonest trigonometry bugs in real code. `math.sin(90)` is `0.894` , not `1` .

13.3. The unit circle: trigonometry for every angle

13.3.1. The problem

SOH-CAH-TOA needs a right triangle, so it only works for angles between 0° and 90°. But a wheel turns through 120°, 270° and 1,000°, and a robot arm can rotate by $-45°$. What is $\sin 150°$? What is $\cos(-30°)$? We need definitions that work for any angle and agree with the triangle definitions where those apply.

13.3.2. Intuition: follow a point round a circle

Draw a circle of radius 1 centred at the origin, the **unit circle**. Start at the point $(1, 0)$ and move anticlockwise around the circle through an angle θ. Call where you end up P.

For an acute angle, drop a perpendicular from P to the x-axis. That gives a right triangle with hypotenuse 1, so the adjacent side (the x-coordinate of P) is $\cos\theta$ and the opposite side (the y-coordinate) is $\sin\theta$. So for acute angles,

$$P = (\cos\theta,\ \sin\theta).$$

Now turn this around and make it the **definition** for every angle.



Unit-circle definition of sine and cosine

Rotate the point $(1, 0)$ anticlockwise about the origin by angle θ (clockwise if θ is negative). The point lands at P. Then

$$\cos\theta = x\text{-coordinate of } P, \qquad \sin\theta = y\text{-coordinate of } P, \qquad \tan\theta = \frac{\sin\theta}{\cos\theta} = \text{slope of } OP.$$

For an angle θ in standard position, the **initial side** is the positive x-axis and the **terminal side** is the ray OP. The four **quadrants** are numbered I to IV anticlockwise, starting from the top right.

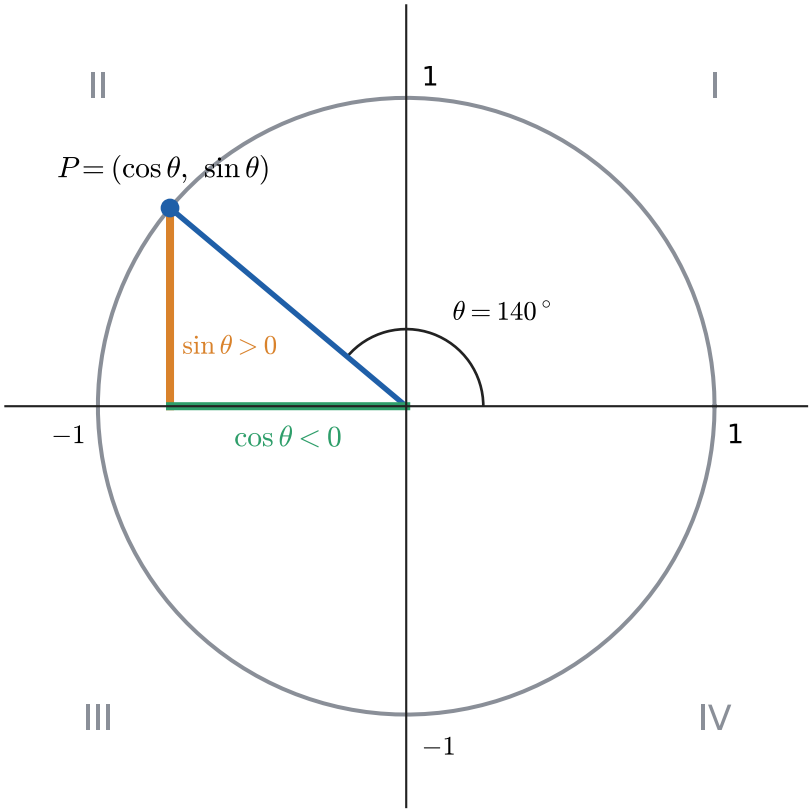


Figure 104. At $\theta = 140°$, P is in quadrant II: its x-coordinate (cosine) is negative and its y-coordinate (sine) is positive.

13.3.3. What the definition gives you immediately

Signs by quadrant. The signs of x and y in each quadrant give:

Quadrant	Angles	sin	cos	tan
I	0° to 90°	+	+	+
II	90° to 180°	+	−	−
III	180° to 270°	−	−	+
IV	270° to 360°	−	+	−

(A traditional mnemonic, anticlockwise from quadrant I: "*A*ll *S*tudents *T*ake *C*alculus". All three are positive in I, Sine in II, Tangent in III, Cosine in IV.)

Bounds. P is on a circle of radius 1, so $-1 \le \sin\theta \le 1$ and $-1 \le \cos\theta \le 1$.

Periodicity. Adding a full turn brings you back to the same point:

$$\sin(\theta + 2\pi) = \sin\theta, \qquad \cos(\theta + 2\pi) = \cos\theta.$$

Sine and cosine are **periodic** with period 2π. Tangent repeats every π, because P and the opposite point $-P$ give the same slope.

Symmetry. Reflecting in the x-axis turns angle θ into $-\theta$. That flips y and leaves x unchanged:

$$\cos(-\theta) = \cos\theta \ \text{(cosine is even)}, \qquad \sin(-\theta) = -\sin\theta \ \text{(sine is odd)}.$$

13.3.4. Reference angles: the method

Example 73. Procedure — Evaluating a trig function at any angle

Input: an angle θ (degrees or radians) and a function (sin, cos or tan).

Steps:

1. Add or subtract multiples of 360° (2π) until the angle is between 0° and 360°.
2. Find the quadrant of the terminal side.
3. Find the **reference angle** α: the acute angle between the terminal side and the x-axis.
   - Quadrant I: $\alpha = \theta$. II: $\alpha = 180° - \theta$. III: $\alpha = \theta - 180°$. IV: $\alpha = 360° - \theta$.
4. Evaluate the function at α (special-angle table or calculator).
5. Attach the sign that function has in that quadrant.

Output: the exact or decimal value.

13.3.5. Worked examples

$\sin 150°$. Quadrant II, reference angle $180° - 150° = 30°$, sine is positive in II:
$\sin 150° = +\sin 30° = \frac{1}{2}$. *Try it:* Evaluate sin 150° step by step .

$\cos 240°$. Quadrant III, reference angle $240° - 180° = 60°$, cosine is negative in III:
$\cos 240° = -\cos 60° = -\frac{1}{2}$. *Try it:* Evaluate cos 240° step by step .

$\tan\left(\frac{11\pi}{6}\right)$. $\frac{11\pi}{6} = 330°$: quadrant IV, reference angle 30°, tangent is negative in IV:
$\tan 330° = -\tan 30° = -\frac{\sqrt{3}}{3}$. *Try it:* Evaluate tan(11π/6) step by step .

$\sin 765°$. $765 - 2 \times 360 = 45$, so $\sin 765° = \sin 45° = \frac{\sqrt{2}}{2}$. *Try it:* Evaluate sin 765° step by step .

13.3.6. Solving trig equations: many answers

Because the functions repeat, an equation like $\sin x = \frac{1}{2}$ has **infinitely many** solutions. Your calculator's $\sin^{-1}$ returns just one of them.

Solve $\sin x = \frac{1}{2}$ for $0 \le x < 2\pi$: *Try it:* Find every solution of sin x = ½ .

$$x_1 = \sin^{-1}\tfrac{1}{2} = \tfrac{\pi}{6} \qquad \text{(principal value, quadrant I)}$$
$$x_2 = \pi - \tfrac{\pi}{6} = \tfrac{5\pi}{6} \qquad \text{(sine is also positive in II, same reference angle)}.$$

The general solution is $x = \frac{\pi}{6} + 2\pi k$ or $x = \frac{5\pi}{6} + 2\pi k$ for any integer k.



The inverse gives one answer, not all of them

$\sin^{-1}$, $\cos^{-1}$ and $\tan^{-1}$ return a single principal value. If the problem has a range, such as "all angles in a full turn", you have to find the others yourself using symmetry: for sine, $\pi - x$; for cosine, $-x$ (or $2\pi - x$); for tangent, $x + \pi$. Forgetting the second solution is the commonest error in trig equations. In code it's the reason `atan2` exists (Section 13.9).



Why "tangent" and "secant"?

Draw the vertical line $x = 1$, which **touches** (is tangent to) the unit circle at $(1, 0)$. Extend the ray OP until it meets that line. The height where it meets is exactly $\tan\theta$, and the length of the extended ray from O (a line that **cuts** the circle, a secant) is exactly $\sec\theta$. The names describe a picture, and the picture also explains why $\tan 90°$ is undefined: at 90° the ray runs parallel to the tangent line and never meets it.



Spin the unit circle

Drag a point around the unit circle. Show its coordinates $(\cos\theta, \sin\theta)$ live, colour the quadrant, and draw the reference angle. Next to it, trace $\sin\theta$ and $\cos\theta$ as graphs against θ as the point moves, so the reader sees the circle "unwrap" into waves. Add a toggle between degrees and radians.

Try it: Unit circle unwrapped · Spin the unit circle

13.4. Graphs of sine, cosine and tangent

13.4.1. From circle to wave

Plot the y-coordinate of the moving point against the angle it has turned through, and you get the sine wave. Plot the x-coordinate and you get the cosine wave, which is the same shape shifted left by a quarter turn: $\cos x = \sin\left(x + \frac{\pi}{2}\right)$.

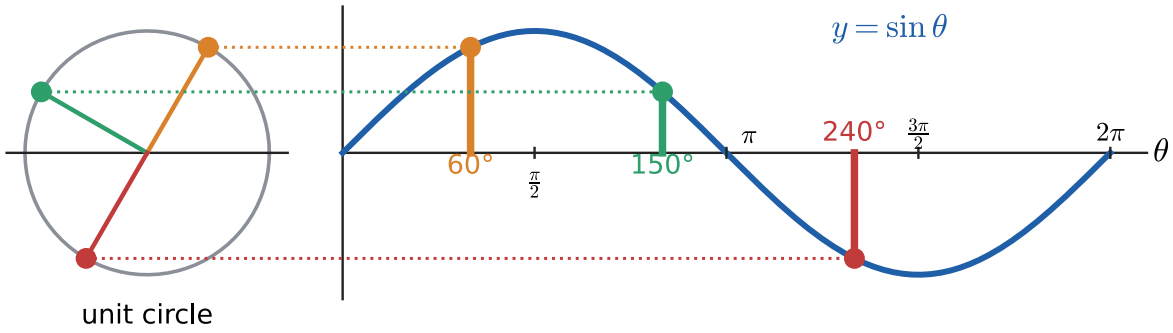


Figure 105. Unwrapping the circle. Each point on the unit circle (60°, 150°, 240°) has its height carried across to the graph at the matching angle. The sine graph is the height of the point plotted against the angle turned.

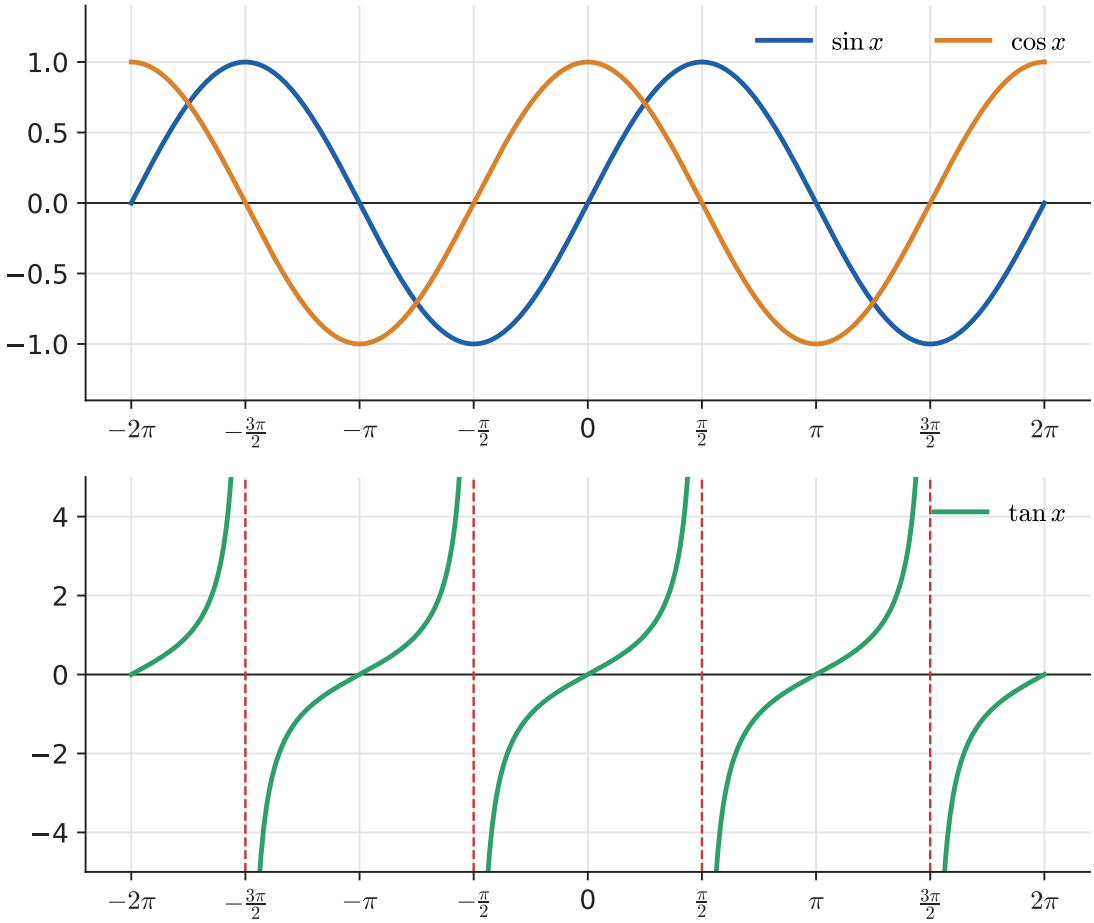


Figure 106. Top: $\sin x$ and $\cos x$, each with period 2π and values between -1 and 1. Bottom: $\tan x$, period π, with vertical asymptotes (dashed) where $\cos x = 0$.

Key features:

- $\sin x$: zeros at $x = k\pi$, maximum 1 at $\frac{\pi}{2} + 2k\pi$, minimum −1 at $\frac{3\pi}{2} + 2k\pi$. Domain all reals, range $[-1, 1]$.
- $\cos x$: the same shape, starting at its maximum: $\cos 0 = 1$.
- $\tan x$: zeros where $\sin x = 0$. It is undefined and has **vertical asymptotes** where $\cos x = 0$, at $x = \frac{\pi}{2} + k\pi$. Range is all reals.

13.4.2. Amplitude, period, phase and midline

Real oscillations don't swing exactly between −1 and 1 with period 2π. You get the general **sinusoid** by applying the function transformations from Chapter 11:

$$y = A\sin\big(B(x - C)\big) + D.$$

Sinusoid parameters

For $y = A\sin(B(x - C)) + D$ (with $A, B > 0$):

- **Amplitude A**: the height of a peak above the midline, half the peak-to-trough distance. (Vertical stretch.)
- **Period $\dfrac{2\pi}{B}$**: the length of one full cycle. (A horizontal squeeze by factor B.) The **frequency** is $\dfrac{B}{2\pi}$ cycles per unit, the reciprocal of the period. B itself is the **angular frequency**, usually written ω (omega), in radians per unit.
- **Phase shift C**: the graph moves right by C. (Horizontal translation.)
- **Midline $y = D$**: the level the wave oscillates about. (Vertical translation.)

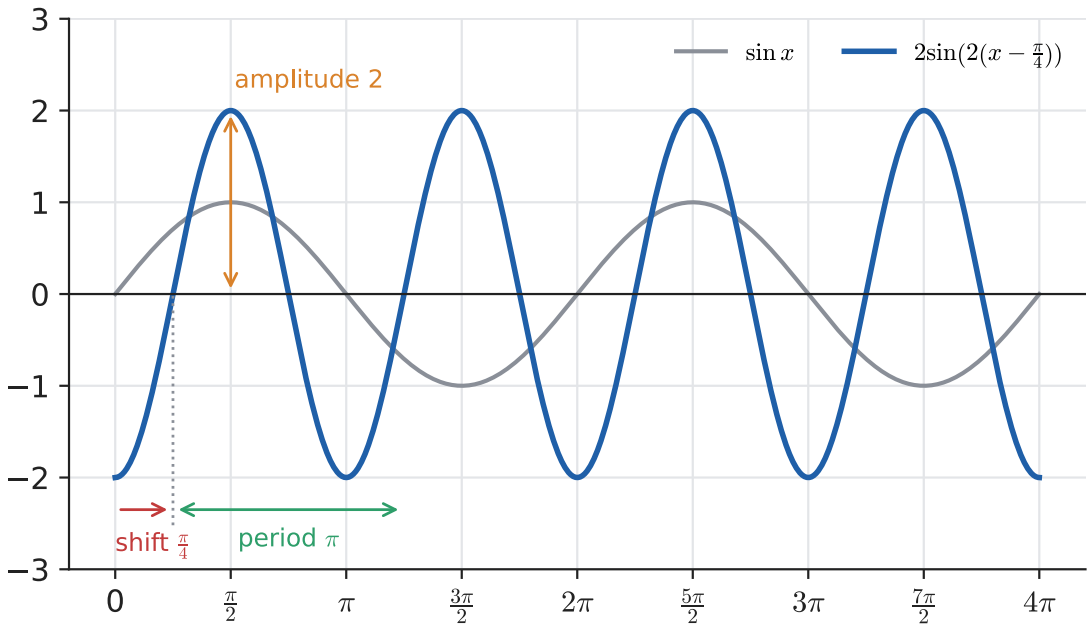


Figure 107. $y = 2\sin(2(x - \frac{\pi}{4}))$ compared with $\sin x$: amplitude 2, period π, shifted right by $\frac{\pi}{4}$.

Factor out B before reading the shift

In $y = \sin(2x - \frac{\pi}{2})$ the phase shift is **not** $\frac{\pi}{2}$. Factor first: $\sin\big(2(x - \frac{\pi}{4})\big)$, so the shift is $\frac{\pi}{4}$. Physicists often write $\sin(\omega t + \varphi)$ and call φ (phi) the **phase**. The actual time shift is then $-\varphi/\omega$. Check which convention a formula is using.

Try it: Adjust amplitude, period and phase .

13.4.3. Worked example: modelling daylight

In a city at about 40° N, the longest day (around 21 June, day 172) has about 15 hours of daylight and the shortest (around 21 December, day 355) about 9 hours. Model daylight hours H against day of the year t with a sinusoid. *Try it:* Fit the daylight model .

$$D = \frac{15+9}{2} = 12 \qquad \text{(midline: average of max and min)}$$

$$A = \frac{15-9}{2} = 3 \qquad \text{(amplitude: half the range)}$$

$$B = \frac{2\pi}{365} \qquad \text{(period one year)}$$

C : the sine rises through the midline a quarter-period before its peak:

$$C = 172 - \frac{365}{4} = 172 - 91.25 = 80.75 \qquad \text{(about 21 March, the equinox).}$$

So $H(t) = 3\sin\left(\frac{2\pi}{365}(t - 80.75)\right) + 12$. Check day 172: the argument is $\frac{2\pi}{365} \times 91.25 = \frac{\pi}{2}$, so $H = 3 + 12 = 15$. Correct.

Example 74. Procedure — Writing a sinusoid from data

Input: the maximum and minimum values, the period, and the location of one maximum (or a rising midline crossing).

Steps:

1. Midline $D = (\text{max} + \text{min})/2$.
2. Amplitude $A = (\text{max} - \text{min})/2$.
3. $B = 2\pi/\text{period}$.
4. Phase shift C = location of a rising midline crossing (for sine), or of a maximum (if you use cosine instead).
5. Write $y = A\sin(B(x - C)) + D$ and check one known point.

Output: a sinusoidal model.

13.4.4. Where you'd use this

Anything that repeats on a schedule can be modelled, at least roughly, by a sinusoid, and the four parameters answer practical questions.

- **Boating (everyday):** at a harbour, high water is 4.2 m at 03:00 and low water 0.6 m, and the tide repeats every 12.4 hours. Using cosine (which peaks at its shift), $h(t) = 1.8\cos\left(\frac{2\pi}{12.4}(t - 3)\right) + 2.4$. A boat that needs 2.0 m of water to cross the harbour bar can cross while $\cos(\dots) \geq (2.0 - 2.4)/1.8 = -0.222$, that is, while $|t - 3| \leq \frac{12.4}{2\pi}\cos^{-1}(-0.222) \approx 3.54$ hours. The window is roughly 23:27 to 06:33. Sailors' "rule of twelfths" is a hand approximation to this sine curve. *Try it:* Find the tide window .

- **Electricity (engineering):** UK mains is 230 V at 50 Hz, but 230 V is an average-power figure (see Section 13.5). The actual voltage is $V(t) = 325\sin(2\pi \cdot 50\,t) = 325\sin(100\pi t)$ volts: amplitude 325 V, period $1/50$ s = 20 ms. US mains is $170\sin(120\pi t)$, which is 120 V at 60 Hz. *Try it:* Plot the mains voltage .
- **Computing:** web traffic, CPU load and electricity demand all have a strong daily cycle. Capacity planners fit $A\sin(\frac{2\pi}{24}(t - C)) + D$ to hourly data. D is the average load, A tells you how much headroom the peak needs, and C tells you when to schedule batch jobs, in the trough. Forecasting libraries such as Prophet model seasonality as a sum of sines and cosines of this kind.

US curriculum

The unit circle, radians, graphs of trig functions, amplitude, period and phase are core **Algebra II** and **Precalculus** topics (grades 10–12). AP Precalculus devotes a whole unit to trigonometric and polar functions.

13.5. Identities: Pythagorean, angle sum, double angle

13.5.1. The problem

In a physics problem you get $\sin\theta = 0.6$ and need $\cos\theta$. A graphics routine needs the sine of a sum of two rotations. A signal-processing derivation needs $\sin^2 x$ written without the square. These all call for **identities**, equations that hold for every angle. They let you rewrite one trig expression as another.



Identity

An **identity** is an equation that holds for **every** value of the variable for which both sides are defined, such as $(a+b)^2 = a^2 + 2ab + b^2$. An ordinary **equation** such as $\sin x = \frac{1}{2}$ holds only for particular values. Some books write $\equiv$ for identities.

13.5.2. The Pythagorean identity, from the unit circle

The point $P = (\cos\theta, \sin\theta)$ lies on the unit circle $x^2 + y^2 = 1$ (Chapter 10). Substitute:

$$\cos^2\theta + \sin^2\theta = 1.$$

That's Pythagoras' theorem for a right triangle with hypotenuse 1, and the unit-circle definition makes it hold for **every** angle, not just acute ones. Dividing through by $\cos^2\theta$ or $\sin^2\theta$ gives two variants:

$$1 + \tan^2\theta = \sec^2\theta, \qquad \cot^2\theta + 1 = \csc^2\theta.$$

Worked example. $\sin\theta = 0.6$ and θ is in quadrant II. Find $\cos\theta$ and $\tan\theta$.

$$\begin{aligned}
\cos^2\theta &= 1 - \sin^2\theta = 1 - 0.36 = 0.64 && \text{(Pythagorean identity)}\\
\cos\theta &= \pm 0.8 && \text{(square root: two candidates)}\\
\cos\theta &= -0.8 && \text{(cosine is negative in quadrant II)}\\
\tan\theta &= \frac{0.6}{-0.8} = -0.75.
\end{aligned}$$

13.5.3. Angle-sum formulas, derived geometrically

Is $\sin(\alpha+\beta) = \sin\alpha + \sin\beta$? Try $\alpha = \beta = 90°$: the left side is $\sin 180° = 0$, the right side is $1 + 1 = 2$. No. So what **is** it?

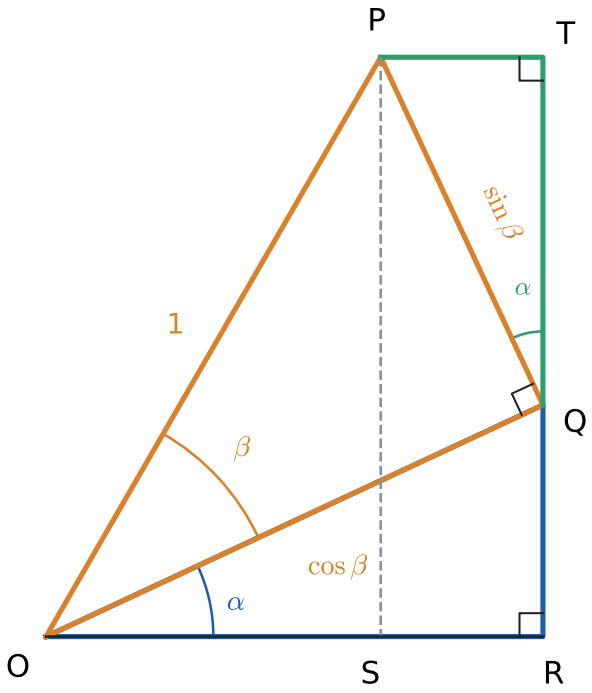


Figure 108. Stacking angle β on top of angle α. The hypotenuse $OP = 1$, so $P = (\cos(\alpha+\beta), \sin(\alpha+\beta))$.

The construction in the figure:

1. Draw the angle α at O from the horizontal, then the angle β on top of it, giving a ray at angle $\alpha+\beta$. Mark P on that ray with $OP = 1$.
2. Drop a perpendicular from P onto the α-ray, meeting it at Q. Triangle OQP has a right angle at Q and hypotenuse 1, so $OQ = \cos\beta$ and $QP = \sin\beta$.
3. Drop a perpendicular from Q to the horizontal at R. Triangle ORQ has hypotenuse $\cos\beta$ and angle α, so $OR = \cos\alpha\cos\beta$ and $RQ = \sin\alpha\cos\beta$.
4. Draw a vertical line up from Q and a horizontal line across from P. They meet at T. In triangle QTP, the angle at Q is α: QT is vertical, and QP is perpendicular to OQ, so the angle QP makes with the vertical equals the angle OQ makes with the horizontal. With hypotenuse $QP = \sin\beta$, we get $QT = \cos\alpha\sin\beta$ and $TP = \sin\alpha\sin\beta$.

Now read off P's coordinates two ways:

$$\begin{aligned}\sin(\alpha+\beta) &= \text{height of } P = RQ + QT = \sin\alpha\cos\beta + \cos\alpha\sin\beta,\\ \cos(\alpha+\beta) &= \text{horizontal position of } P = OR - TP = \cos\alpha\cos\beta - \sin\alpha\sin\beta.\end{aligned}$$

The picture assumes acute angles with an acute sum, but the formulas hold for all angles. (The cleanest general proof uses rotation matrices, Volume 2, Chapter 27 , or complex numbers, Chapter 15.) Replacing β with $-\beta$ and using even/odd symmetry gives the difference formulas.



Angle-sum and difference formulas

$$\begin{aligned}\sin(\alpha \pm \beta) &= \sin\alpha\cos\beta \pm \cos\alpha\sin\beta \\ \cos(\alpha \pm \beta) &= \cos\alpha\cos\beta \mp \sin\alpha\sin\beta \\ \tan(\alpha \pm \beta) &= \frac{\tan\alpha \pm \tan\beta}{1 \mp \tan\alpha\tan\beta}\end{aligned}$$

$\pm$ means "plus or minus" and $\mp$ means "the opposite sign": take the top signs together or the bottom signs together. The cosine formula **flips** the sign.

Worked example: an exact value for $\sin 75^\circ$.

$$\begin{aligned}\sin 75^\circ &= \sin(45^\circ + 30^\circ) \\ &= \sin 45^\circ\cos 30^\circ + \cos 45^\circ\sin 30^\circ && \text{(angle sum)} \\ &= \frac{\sqrt{2}}{2}\cdot\frac{\sqrt{3}}{2} + \frac{\sqrt{2}}{2}\cdot\frac{1}{2} && \text{(special angles)} \\ &= \frac{\sqrt{6}+\sqrt{2}}{4} \approx \frac{2.449490 + 1.414214}{4} = 0.965926.\end{aligned}$$

13.5.4. Double-angle formulas

Put $\beta = \alpha$ in the sum formulas:

$$\begin{aligned}\sin 2\alpha &= \sin\alpha\cos\alpha + \cos\alpha\sin\alpha = 2\sin\alpha\cos\alpha, \\ \cos 2\alpha &= \cos^2\alpha - \sin^2\alpha.\end{aligned}$$

Using $\sin^2\alpha + \cos^2\alpha = 1$ to remove one or the other gives two more forms of $\cos 2\alpha$:

$$\cos 2\alpha = 2\cos^2\alpha - 1 = 1 - 2\sin^2\alpha.$$

Rearranged, these are the **power-reduction** formulas that calculus uses all the time (Volume 2, Chapter 19):

$$\sin^2\alpha = \frac{1-\cos 2\alpha}{2}, \qquad \cos^2\alpha = \frac{1+\cos 2\alpha}{2}.$$

Note what they say: the square of a wave at frequency f is a constant plus a wave at frequency $2f$. That fact explains why the **power** of an AC signal pulses at twice the mains frequency.

13.5.5. Choosing an approach

- **Need one ratio given another?** Use the Pythagorean identity, then fix the sign by quadrant.
- **Exact value at an awkward angle** (15°, 75°, 105°)? Write it as a sum or difference of special angles.
- **Squared trig function to integrate or average?** Use power reduction.

- **Proving an identity?** Work on the more complicated side, convert everything to sin and cos, and look for $\sin^2 + \cos^2$. Never "cross over" between the sides as if solving an equation, because that assumes what you're trying to prove.
- **Just checking numerically?** Plug in a random angle like 0.7 rad. That disproves a false identity instantly, though it can't prove a true one.



Trig functions are not linear

$\sin(\alpha+\beta) \neq \sin\alpha + \sin\beta$, $\sin 2\alpha \neq 2\sin\alpha$, and $\cos(x/2) \neq \frac{1}{2}\cos x$. "sin" is a function name, not a number you can distribute or cancel. $\frac{\sin x}{x}$ does **not** simplify to "sin", and $\frac{\sin 2x}{\sin x}$ is $2\cos x$, not 2.

13.5.6. Where you'd use this

- **Electricity: why "230 volts" is not the peak.** Power in a resistor is V^2/R. With $V = V_0 \sin \omega t$, power reduction gives $V^2 = V_0^2 \cdot \frac{1-\cos 2\omega t}{2}$. The cosine term averages to zero over a cycle, so the **average** power is $\frac{V_0^2}{2R}$: the same as a steady DC voltage of $V_0/\sqrt{2}$. That is the **RMS** (root-mean-square) voltage printed on every appliance. For UK mains, $325/\sqrt{2} \approx 230$ V. The cosine term at 2ω is the power pulsing 100 times a second.
- **Music: tuning by beats.** Add $\sin(\alpha+\beta)$ and $\sin(\alpha-\beta)$ using the angle-sum formulas and the $\cos\alpha\sin\beta$ terms cancel: $\sin(\alpha+\beta) + \sin(\alpha-\beta) = 2\sin\alpha\cos\beta$. Play a 440 Hz string against a 442 Hz tuning reference and you hear a 441 Hz tone whose loudness swells and fades as $\cos(2\pi \cdot 1 \cdot t)$, peaking twice a second. Guitarists tune by turning the peg until those 2 Hz "beats" slow to nothing.
- **Computing: oscillators without calling `sin`.** The angle-sum formula gives $\sin((n+1)\theta) = 2\cos\theta\,\sin(n\theta) - \sin((n-1)\theta)$. (Add the sum and difference formulas for $\sin(n\theta \pm \theta)$.) An audio synthesiser or a graphics loop that needs $\sin$ at evenly spaced angles computes $2\cos\theta$ once, then gets each new value with one multiply and one subtract. The same recurrence generates rotation steps in old games and in DSP chips.



Ptolemy's table of chords

Around 150 CE, Claudius Ptolemy built a table of chord lengths (equivalent to a sine table) in half-degree steps, accurate to about five decimal places. He used, in effect, the angle-sum and half-angle formulas, starting from the exactly known chords for 36°, 60°, 72° and 90° and working out the rest. His **Almagest** was the standard astronomy text for 1,400 years. Your CPU's `sin` uses the same strategy: reduce the argument to a small range using identities, then use a short polynomial approximation (Volume 2, Chapter 32).

13.6. Any triangle: the laws of sines and cosines

13.6.1. The problem

A surveyor on one bank of a river marks two points A and B, 500 m apart. A tree C stands on the far bank. From A, the angle between B and the tree is 52°. From B, the angle between A and the tree is 71°. How far is the tree from each point, and how wide is the river?

No right angle appears anywhere, so SOH-CAH-TOA doesn't apply directly. We need tools for **any** triangle.



Labelling convention

In triangle ABC, the angles at the vertices are A, B, C, and the side **opposite** each angle gets the matching lower-case letter: side a faces angle A, and so on. The angles always add to $180°$.

13.6.2. The law of sines, derived

Drop the altitude h from C to side AB. It splits the triangle into two right triangles, and in each one SOH gives:

$$\begin{aligned} h &= b\sin A && \text{(left triangle, hypotenuse } b\text{)} \\ h &= a\sin B && \text{(right triangle, hypotenuse } a\text{)} \\ b\sin A &= a\sin B && \text{(both equal } h\text{)} \\ \frac{a}{\sin A} &= \frac{b}{\sin B} && \text{(divide by } \sin A \sin B\text{).} \end{aligned}$$

Dropping an altitude from a different vertex brings c and C in the same way. The result is the **law of sines**:

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}.$$

Each side is proportional to the sine of the angle opposite it. (The common ratio has a meaning: it's the diameter of the triangle's circumscribed circle.)

The river, solved. $C = 180° - 52° - 71° = 57°$, and $c = AB = 500$. *Try it:* Solve the river triangle

.

$$\begin{aligned} AC = b &= \frac{c\sin B}{\sin C} = \frac{500\sin 71°}{\sin 57°} = \frac{500 \times 0.945519}{0.838671} \approx 563.7 \text{ m} \\ BC = a &= \frac{c\sin A}{\sin C} = \frac{500\sin 52°}{\sin 57°} = \frac{500 \times 0.788011}{0.838671} \approx 469.8 \text{ m} \\ \text{width} &= AC\sin 52° = 563.7 \times 0.788011 \approx 444.2 \text{ m} && \text{(perpendicular distance from } C \text{ to line } AB\text{).} \end{aligned}$$

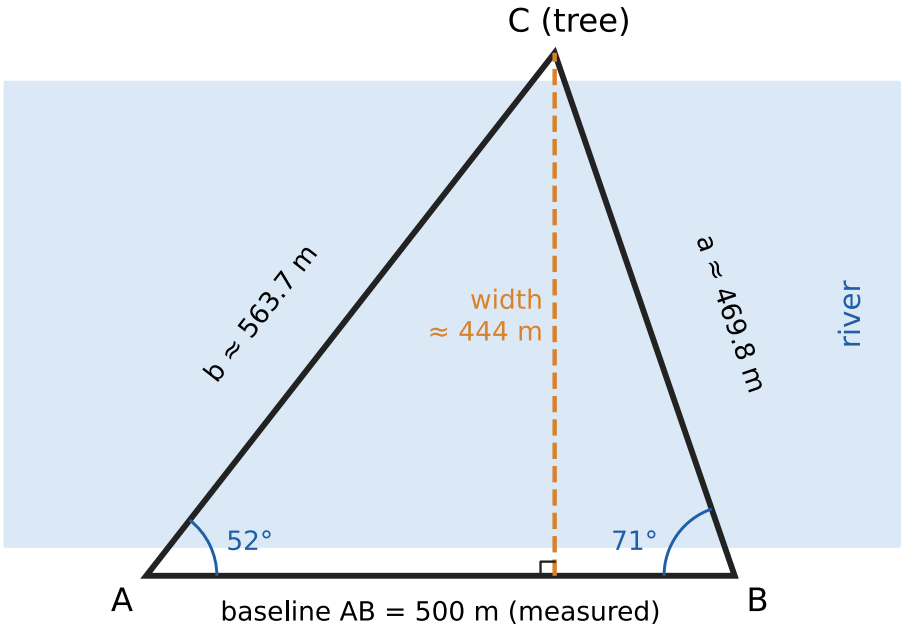


Figure 109. The river problem, solved. Only the baseline AB and the two angles were measured. The law of sines gives the other two sides, and the dashed perpendicular is the river's width.

This is **triangulation**. From one carefully measured baseline and many angle measurements, surveyors mapped whole countries. The 18th–19th-century Great Trigonometrical Survey of India measured the height of Everest this way.

13.6.3. The law of cosines: Pythagoras, generalised

The law of sines needs at least one angle–opposite side pair. If you know two sides and the angle **between** them (or all three sides), you need the law of cosines instead.

Derivation. Use the triangle in the figure. Put A at the origin and B at $(c, 0)$. C is at distance b from A at angle A, so $C = (b\cos A,\ b\sin A)$. Then a is the distance from B to C (distance formula, Chapter 10):

$$\begin{aligned} a^2 &= (b\cos A - c)^2 + (b\sin A - 0)^2 \\ &= b^2\cos^2 A - 2bc\cos A + c^2 + b^2\sin^2 A && \text{(expand)} \\ &= b^2(\cos^2 A + \sin^2 A) + c^2 - 2bc\cos A && \text{(group the } b^2 \text{ terms)} \\ &= b^2 + c^2 - 2bc\cos A && \text{(Pythagorean identity).} \end{aligned}$$

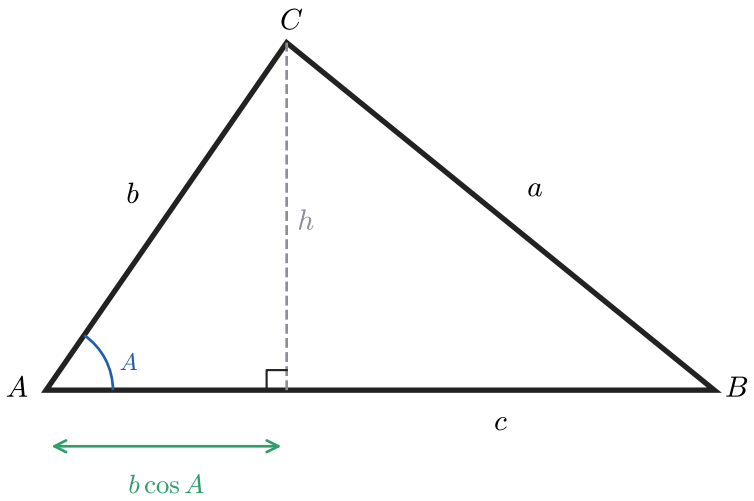


Figure 110. Law of cosines setup: C sits at horizontal distance $b\cos A$ from A.

Laws of sines and cosines

For any triangle:

Sines: $\dfrac{a}{\sin A} = \dfrac{b}{\sin B} = \dfrac{c}{\sin C}$ **Cosines:** $a^2 = b^2 + c^2 - 2bc\cos A$

(and the same with the letters cycled: $b^2 = a^2 + c^2 - 2ac\cos B$, $c^2 = a^2 + b^2 - 2ab\cos C$).

When $A = 90°$, $\cos A = 0$ and the law of cosines **is** Pythagoras. The term $-2bc\cos A$ is the correction for a non-right angle. It is negative (side shorter) for acute A and positive (side longer) for obtuse A.

Worked example: three sides, find an angle. A triangle has sides 5, 7 and 8. Find the angle opposite the side of length 7. *Try it:* Solve the 5-7-8 triangle .

$$\begin{aligned} 7^2 &= 5^2 + 8^2 - 2(5)(8)\cos\theta && \text{(law of cosines; } \theta \text{ opposite side 7)} \\ 49 &= 89 - 80\cos\theta \\ 80\cos\theta &= 40 \\ \cos\theta &= 0.5 \quad\Rightarrow\quad \theta = \cos^{-1}(0.5) = 60°. \end{aligned}$$

Worked example: navigation. A ship sails 20 km on a bearing of 040°, then turns and sails 30 km on a bearing of 110°. How far is it from port? *Try it:* Solve the navigation triangle .

Bearings are measured clockwise from north. The ship's heading changes by $110° - 40° = 70°$, so the **interior** angle of the triangle at the turning point is $180° - 70° = 110°$.

$$\begin{aligned} d^2 &= 20^2 + 30^2 - 2(20)(30)\cos 110° \\ &= 400 + 900 - 1200 \times (-0.342020) && (\cos 110° < 0\text{: obtuse}) \\ &= 1300 + 410.42 = 1710.42 \\ d &\approx 41.36 \text{ km}. \end{aligned}$$

The obtuse angle makes the distance **longer** than $\sqrt{20^2 + 30^2} \approx 36.06$ km, as the sign rule predicts. (In Section 13.9 we solve the same problem with components and atan2, which also gives the bearing back to port.)

13.6.4. Choosing a method

You know	Use
Right triangle, any two pieces	SOH-CAH-TOA and Pythagoras
Two angles and any side (AAS, ASA)	Law of sines (find the third angle from the 180° sum first)
Two sides and the angle between them (SAS)	Law of cosines for the third side
Three sides (SSS)	Law of cosines for an angle (start with the largest angle)
Two sides and an angle **not** between them (SSA)	Law of sines, but check for the **ambiguous case**



The ambiguous case (SSA)

Given $A = 30°, a = 6, b = 10$: the law of sines gives $\sin B = 10 \sin 30° / 6 = 0.8333$. Your calculator says $B \approx 56.44°$, but $180° - 56.44° = 123.56°$ has the same sine. Both fit, since $30 + 123.56 < 180$, so **two different triangles** match the data (with third sides about 11.98 and 5.34). Whenever you use $\sin^{-1}$ to find an angle, ask whether its supplement also works. The law of cosines doesn't have this problem, because $\cos^{-1}$ covers 0° to 180° without ambiguity. That's why SSS starts with cosines and finds the largest angle first.

Try it: See both SSA solutions .

13.6.5. Where you'd use this

- **Sport:** a baseball diamond is a 90 ft square, and the pitcher's mound sits 60.5 ft from home plate on the diagonal towards second base. That diagonal makes 45° with the first-base line, so the throw from the mound to first base is $\sqrt{60.5^2 + 90^2 - 2(60.5)(90)\cos 45°} \approx 63.7$ ft. That's SAS, so it's a law-of-cosines problem.
- **Navigation and location:** a hiker, a ship's navigator or a search-and-rescue team takes compass bearings to two known landmarks and intersects them. That's ASA, and the law of sines gives the distances. Mobile networks and Wi-Fi positioning do the same thing with signal **angles** (and, more often, with distances, which is **trilateration**).

- **Computing:** a 3D mesh stores vertex positions, so every triangle's three side lengths are known. A renderer or physics engine that needs the angle at a vertex (for smooth shading weights or for mesh-quality checks that flag long thin "sliver" triangles) gets it from SSS with the law of cosines. In robotics, a two-link arm with links of 30 cm and 25 cm that must reach a point 40 cm away needs an elbow angle C with $40^2 = 30^2 + 25^2 - 2(30)(25)\cos C$, so $\cos C = (900 + 625 - 1600)/1500 = -0.05$ and $C \approx 92.9°$. That's **inverse kinematics**, and this is its simplest case.



Measuring the Earth

Around 240 BCE, Eratosthenes noticed that at noon on the summer solstice the Sun shone straight down a well in Syene, while in Alexandria, roughly 800 km north, a vertical stick cast a shadow showing the Sun was about 7.2° (one fiftieth of a circle) from vertical. Then $s = r\theta$ gives the Earth's circumference as about $50 \times 800 = 40{,}000$ km, close to the true 40,075 km. That's arc length and radians, more than 2,000 years before the word "radian".

13.7. Polar coordinates

13.7.1. The problem

A radar screen reports a contact "5 km away at 30°". A robot vacuum knows its wheel has turned through some angle at some distance from its axle. Spirals, flower petals and orbits are awkward to describe as $y = f(x)$. All of these are naturally given by **how far** and **which direction**, not by **how far across and how far up**.

13.7.2. Two descriptions of one point



Polar coordinates

A point's **polar coordinates** (r, θ) are its distance r from the origin (the **pole**) and the angle θ from the positive x-axis (the **polar axis**), measured anticlockwise.

$$\begin{aligned} \textbf{Polar} \to \textbf{Cartesian:} &\quad x = r\cos\theta, \quad y = r\sin\theta \\ \textbf{Cartesian} \to \textbf{polar:} &\quad r = \sqrt{x^2 + y^2}, \quad \theta = \operatorname{atan2}(y, x) \end{aligned}$$

The first line is the unit-circle definition scaled by r. In the second, atan2 is the quadrant-aware inverse tangent (Section 13.9). Polar coordinates aren't unique: (r, θ), $(r, \theta + 2\pi)$ and $(-r, \theta + \pi)$ all name the same point.

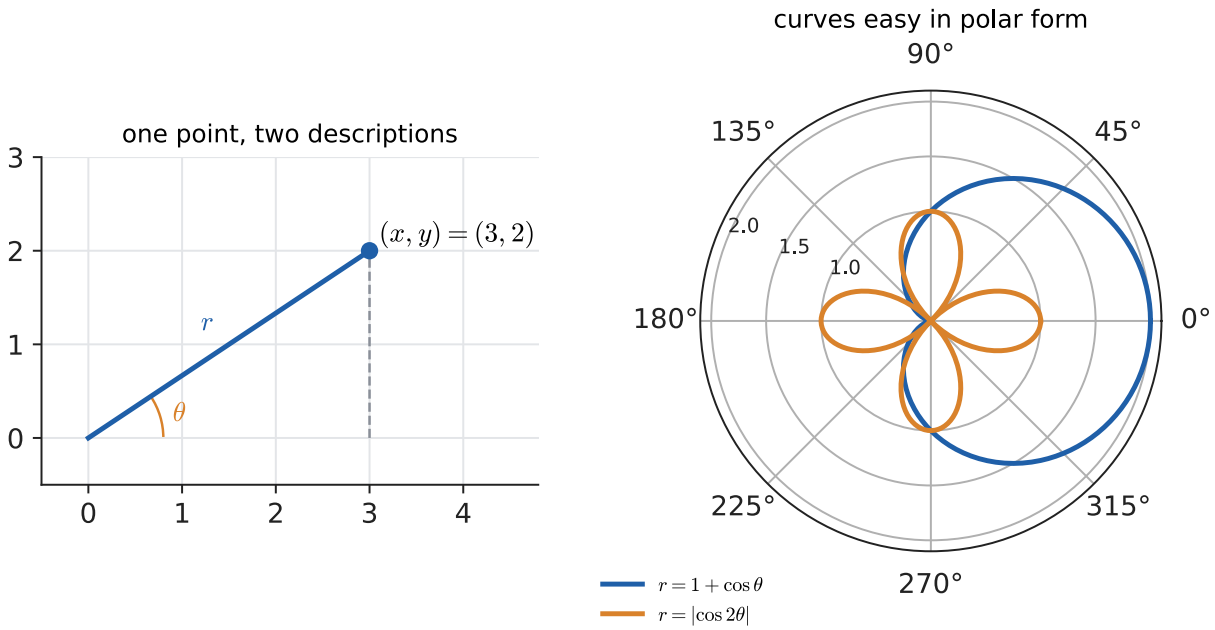


Figure 111. Left: the point (3, 2) is also $r = \sqrt{13}$, $\theta \approx 33.7°$. Right: curves that are simple in polar form, a cardioid and a four-petal rose.

13.7.3. Worked examples

Cartesian to polar: $(3, 2)$.

$$r = \sqrt{3^2 + 2^2} = \sqrt{13} \approx 3.606, \qquad \theta = \tan^{-1}\tfrac{2}{3} \approx 33.69° \ \text{(quadrant I, so } \tan^{-1} \text{ is fine)}.$$

Cartesian to polar: $(-3, 2)$. $r = \sqrt{13}$ again. $\tan^{-1}(2/(-3)) = \tan^{-1}(-0.667) \approx -33.69°$, which points into quadrant IV, the **wrong** quadrant. The point is in quadrant II, so add 180°: $\theta \approx 146.31°$. (atan2 does this correction for you.)

Polar to Cartesian: $r = 4$, $\theta = \frac{5\pi}{6}$. $x = 4\cos\frac{5\pi}{6} = 4 \times (-\frac{\sqrt{3}}{2}) = -2\sqrt{3} \approx -3.464$, $y = 4\sin\frac{5\pi}{6} = 4 \times \frac{1}{2} = 2$.

Curves. $r = 3$ is a circle (compare $x^2 + y^2 = 9$). $r = \theta$ is a spiral. $r = 1 + \cos\theta$ is a heart-shaped **cardioid**, which is also the pickup pattern of a cardioid microphone.

13.7.4. Where you'd use this

Radar and sonar; robot arm kinematics (joint angles and link lengths); orbital mechanics; antenna and microphone directional patterns; and complex numbers, where r and θ become modulus and argument (Chapter 15). Two concrete cases:

- **Games:** a gamepad thumbstick reports (x, y), each between −1 and 1. Pushed to $(0.6, -0.8)$, it gives $r = \sqrt{0.36 + 0.64} = 1$ (full speed) and $\theta = \operatorname{atan2}(-0.8, 0.6) \approx -53.1°$. Games convert to polar so they can apply a **dead zone** (ignore $r < 0.15$, because worn sticks drift) and a response curve on r without bending the direction.
- **Air traffic and radar:** a radar reports an aircraft at 40 km on a bearing of 060°. With east as x and north as y (bearings run clockwise from north), its position is $(40\sin 60°, 40\cos 60°) \approx (34.6, 20.0)$ km. Every radar display does this polar-to-Cartesian

conversion for each blip on every sweep.



Polar sketcher

Type $r = f(\theta)$ (for example $\cos(k\theta)$ with k on a slider) and watch the curve trace out as θ sweeps from 0 to 2π, with the radius line drawn live. Integer k gives roses with k or $2k$ petals. Find the rule.

Try it: Rose curves r = cos(kθ)

13.8. Waves and oscillation

13.8.1. The problem

Pull a mass on a spring down and let go. It bounces up and down with a regular rhythm. A pendulum, a guitar string, the air pressure at your eardrum and the voltage in a wall socket all do something similar. What function describes this motion, and why that one?

13.8.2. Simple harmonic motion

If you studied Mechanics, you met **simple harmonic motion** (SHM): motion in which the restoring force is proportional to the displacement and directed back towards equilibrium. For a spring with stiffness k and mass m, that's Hooke's law with Newton's second law:

$$m\,\ddot{x} = -kx,$$

where $\ddot{x}$ (read "x double-dot") is the acceleration, the second derivative of position with respect to time (Volume 2, Chapter 17). The solution is a sinusoid:

$$x(t) = A\cos(\omega t + \varphi), \qquad \omega = \sqrt{\frac{k}{m}}.$$

Why does a sinusoid appear? Think of the unit circle again. A point moving round a circle at a steady angular speed ω has x-coordinate $\cos(\omega t)$, and its acceleration always points to the centre, so the x-component of the acceleration is proportional to $-x$. **SHM is the shadow of uniform circular motion.** A cylinder on a turntable lit from the side casts a shadow that moves in exactly simple harmonic motion. *Try it:* Watch SHM as a shadow .

Simple harmonic motion

Simple harmonic motion is oscillation described by $x(t) = A\cos(\omega t + \varphi)$.

- A: **amplitude** (maximum displacement).
- ω: **angular frequency**, in radians per second.
- $T = \dfrac{2\pi}{\omega}$: **period**, in seconds per cycle.
- $f = \dfrac{1}{T} = \dfrac{\omega}{2\pi}$: **frequency**, in cycles per second (hertz, Hz).
- φ: **phase constant** (where in the cycle the motion starts).

The period of SHM doesn't depend on the amplitude. A small bounce and a big bounce take the same time.

Worked example. A 0.5 kg mass hangs on a spring with $k = 200$ N/m. It is pulled 4 cm below equilibrium and released. *Try it:* Run the spring example .

$$\omega = \sqrt{k/m} = \sqrt{200/0.5} = \sqrt{400} = 20 \text{ rad/s}$$

$$T = \frac{2\pi}{\omega} = \frac{2\pi}{20} \approx 0.314 \text{ s}, \qquad f = \frac{1}{T} \approx 3.18 \text{ Hz}$$

$x(t) = 0.04\cos(20t)$ m (starts at max displacement, so $\varphi = 0$, measuring down as positive).

The pendulum. For a pendulum of length L, the restoring force is proportional to $\sin\theta$, not θ, so the motion isn't exactly SHM. For small swings, though, $\sin\theta \approx \theta$ (in radians, Section 13.2), and the motion becomes SHM with $\omega = \sqrt{g/L}$, giving period $T = 2\pi\sqrt{L/g}$. A 1 m pendulum has $T = 2\pi\sqrt{1/9.81} \approx 2.006$ s, which is why a "seconds pendulum" (one second per swing) is about a metre long.

13.8.3. Waves

A **wave** is an oscillation that travels. A sinusoidal wave moving along a string is

$$y(x, t) = A\sin(kx - \omega t),$$

where $k = 2\pi/\lambda$ is the **wavenumber** (λ, lambda, is the **wavelength**) and the wave travels at speed $v = \omega/k = f\lambda$. Sound in air at 343 m/s with frequency 440 Hz (concert A) has wavelength $343/440 \approx 0.78$ m.

13.8.4. Sound, and adding sines

A pure sine wave sounds like a tuning fork: clean and rather dull. Real instruments produce a **fundamental** frequency plus **harmonics** at whole-number multiples of it, each with its own amplitude. The mix of harmonics is what makes a violin sound different from a clarinet at the same pitch. *Try it:* Spectrum of three sines .

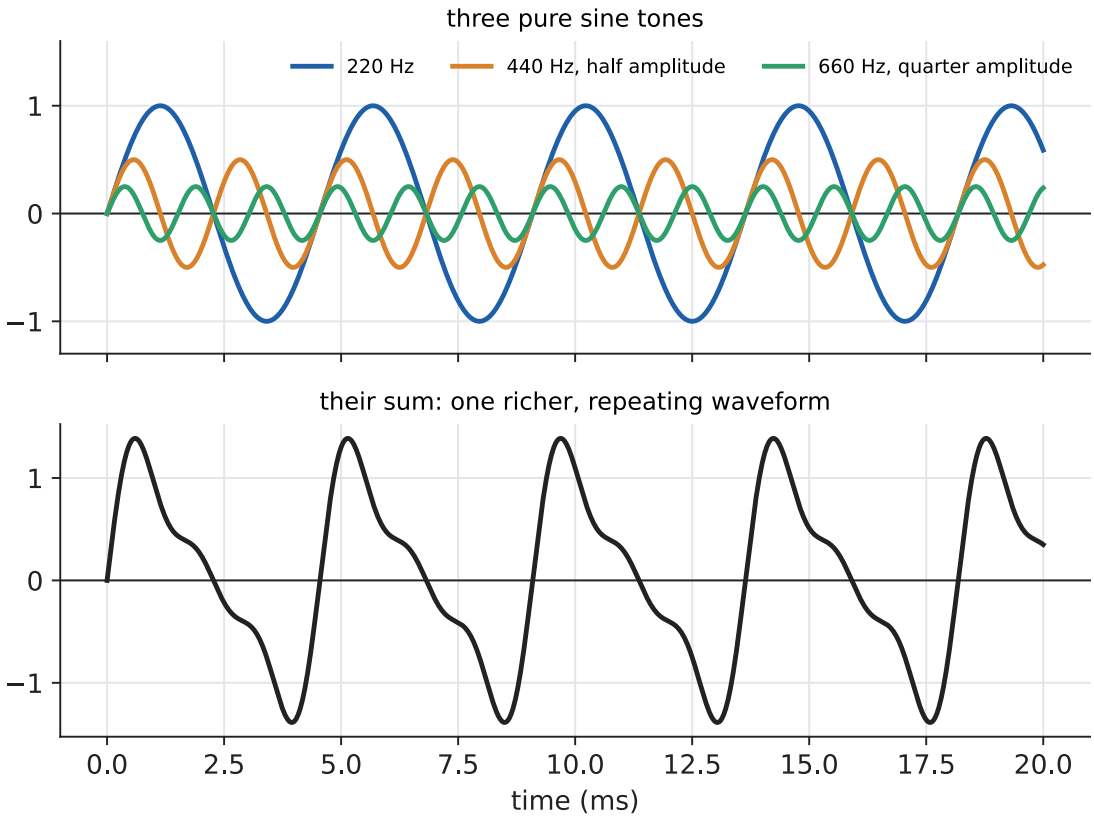


Figure 112. Three sine tones (220, 440 and 660 Hz, with decreasing amplitude) and their sum, a single periodic waveform with period 1/220 s.

13.8.5. Where you'd use this

- **Noise-cancelling headphones:** a microphone picks up the engine drone, say $0.2\sin(2\pi \cdot 120\,t)$, and the headphones play the same wave shifted by half a period, $0.2\sin(2\pi \cdot 120\,t + \pi) = -0.2\sin(2\pi \cdot 120\,t)$. The two add to zero at your ear. Steady low hums cancel well; sudden sounds don't, because the electronics can't predict them in time.
- **Earthquakes and buildings:** a building sways like a mass on a spring with its own natural frequency $f = \omega/2\pi$. If ground shaking arrives near that frequency, each push lands in step with the swing and the amplitude grows (**resonance**). Engineers estimate a building's natural period at roughly 0.1 s per storey, so a 10-storey block (about 1 s) is most at risk from shaking at about 1 Hz.
- **Computing:** Wi-Fi and 4G encode bits by changing the amplitude and phase of sine-wave carriers, so each received symbol is decoded by asking "what are A and φ for this short burst?". The same parameters from Section 13.4 carry your data.



Fourier's big idea

In 1807 Joseph Fourier claimed that **any** periodic signal, even a square wave with sharp corners, can be written as a sum of sines and cosines at multiples of a base frequency. His contemporaries, Lagrange among them, were sceptical. The claim turned out (with careful conditions) to be true, and **Fourier analysis** is now the basis of audio compression (MP3, AAC), image compression (JPEG uses a cosine transform), MRI, radio, spectrum analysers and fast polynomial multiplication. The Fast Fourier Transform (FFT, 1965) is often listed among the most important algorithms of the 20th century.

13.9. Trigonometry in computing

13.9.1. Rotating a point in 2D

The problem. A game sprite at (x, y) relative to its pivot has to rotate by angle θ. Where does it end up?

Derivation. Write the point in polar form: $x = r\cos\varphi$, $y = r\sin\varphi$, with r its distance from the origin and φ its current angle. Rotating keeps r the same and adds θ to the angle, so the new point is

$$
\begin{aligned}
x' &= r\cos(\varphi + \theta) \\
&= r(\cos\varphi\cos\theta - \sin\varphi\sin\theta) && \text{(angle-sum formula)} \\
&= (r\cos\varphi)\cos\theta - (r\sin\varphi)\sin\theta \\
&= x\cos\theta - y\sin\theta, \\
y' &= r\sin(\varphi + \theta) \\
&= r(\sin\varphi\cos\theta + \cos\varphi\sin\theta) && \text{(angle-sum formula)} \\
&= x\sin\theta + y\cos\theta.
\end{aligned}
$$

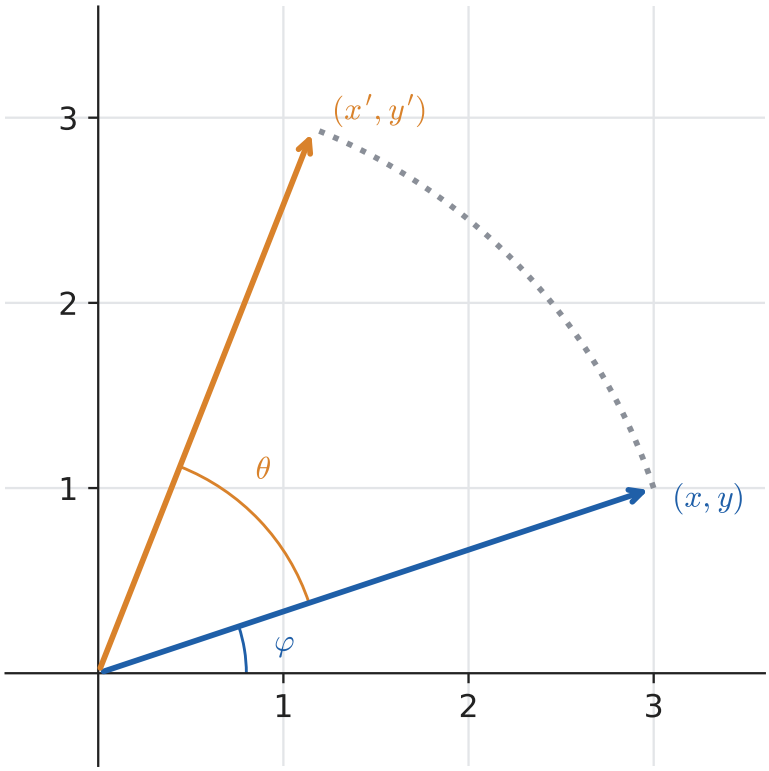


Figure 113. Rotation by θ keeps the distance from the origin fixed and adds θ to the polar angle φ.

2D rotation formula

Rotating (x, y) anticlockwise about the origin by θ:

$$x' = x\cos\theta - y\sin\theta, \qquad y' = x\sin\theta + y\cos\theta.$$

In Volume 2, Chapter 27 this becomes multiplication by the **rotation matrix** $\begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}$. To rotate about some other point (c_x, c_y), translate it to the origin, rotate, and translate back.

Worked example. Rotate $(2, 0)$ by 30°: $x' = 2\cos 30° - 0 = \sqrt{3} \approx 1.732$, $y' = 2\sin 30° + 0 = 1$. The distance is still $\sqrt{3+1} = 2$. Correct.

Worked example. Rotate $(3, 1)$ by 90°: $\cos 90° = 0$, $\sin 90° = 1$, so $x' = -1$, $y' = 3$. Rotating by 90° sends $(x, y) \mapsto (-y, x)$, a trick worth remembering for perpendicular vectors.

```python
import math

def rotate(x, y, theta, cx=0.0, cy=0.0):
    """Rotate (x, y) anticlockwise by theta radians about (cx, cy)."""
    c, s = math.cos(theta), math.sin(theta)
    dx, dy = x - cx, y - cy                # translate pivot to origin
    return (cx + dx * c - dy * s,          # rotate, translate back
            cy + dx * s + dy * c)

print(rotate(2, 0, math.radians(30)))
print(rotate(3, 1, math.radians(90)))
print(rotate(3, 1, math.radians(50)))
print(rotate(5, 5, math.pi, cx=4, cy=4))   # half-turn about (4, 4)
```

```
(1.7320508075688774, 0.9999999999999999)
(-0.9999999999999998, 3.0)
(1.1623183859406403, 2.9409209390434734)
(3.0, 3.0)
```

Notice `0.9999999999999999` and `-0.9999999999999998`. Neither $\pi/6$ nor $\pi/2$ is exactly representable as a float, so `math.cos(math.radians(90))` is about 6×10^{-17}, not 0. Compare floating-point trig results with a tolerance (`math.isclose`), never with `==`.

13.9.2. atan2: why it exists

The problem. An enemy at (e_x, e_y) has to turn to face the player at (p_x, p_y). The direction vector is $(d_x, d_y) = (p_x - e_x, p_y - e_y)$, and the facing angle θ satisfies $\tan\theta = d_y/d_x$. So is the answer `atan(dy/dx)`? *Try it:* Watch atan aim the wrong way.

No, for two reasons:

1. **Lost quadrant.** $(2, 1)$ and $(-2, -1)$ have the same ratio $1/2$, so `atan` returns 26.6° for both. One of them actually points the opposite way (−153.4°). The division throws away the signs.
2. **Division by zero.** Straight up or down, $d_x = 0$.

`atan2(y, x)` takes the two components **separately**. It uses their signs to pick the correct quadrant and returns an angle in $(-\pi, \pi]$ (−180° to 180°). It handles $x = 0$ without trouble. Note the argument order: **y first**, matching $\tan^{-1}(y/x)$.

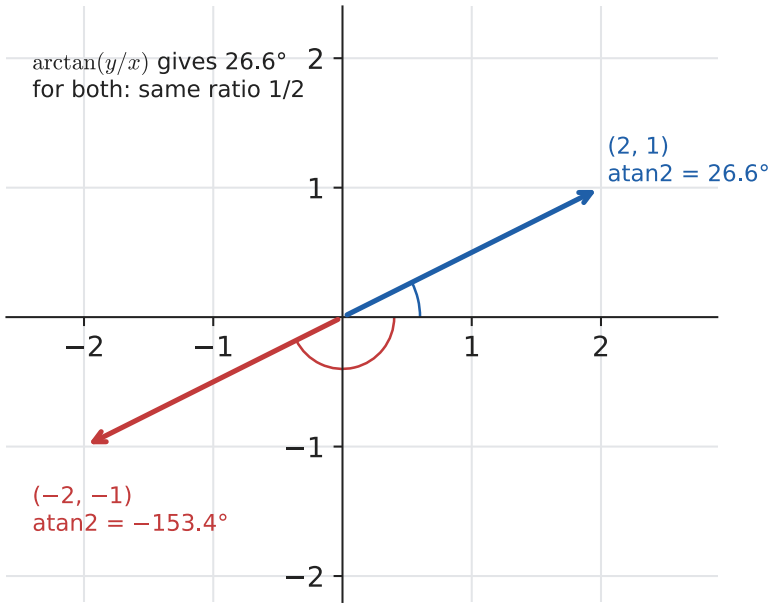


Figure 114. Two direction vectors with the same slope. atan(y/x) can't tell them apart, but atan2(y, x) can.

```python
import math

for x, y in [(2, 1), (-2, -1), (-2, 1), (0, 3), (0, -3)]:
    naive = math.degrees(math.atan(y / x)) if x != 0 else float("nan")
    good = math.degrees(math.atan2(y, x))
    print(f"({x:>2}, {y:>2})  atan(y/x) = {naive:8.2f}   atan2(y, x) = {good:8.2f}")
```

```
( 2,  1)  atan(y/x) =    26.57   atan2(y, x) =    26.57
(-2, -1)  atan(y/x) =    26.57   atan2(y, x) =  -153.43
(-2,  1)  atan(y/x) =   -26.57   atan2(y, x) =   153.43
( 0,  3)  atan(y/x) =      nan   atan2(y, x) =    90.00
( 0, -3)  atan(y/x) =      nan   atan2(y, x) =   -90.00
```

The navigation problem by components. Let east be x and north be y. A bearing β (clockwise from north) points in the direction $(\sin\beta, \cos\beta)$. Note that this is the reverse of the usual math convention.

$$\begin{aligned}
&\text{leg 1: } 20(\sin 40^\circ, \cos 40^\circ) = (12.856,\ 15.321)\\
&\text{leg 2: } 30(\sin 110^\circ, \cos 110^\circ) = (28.191,\ -10.261)\\
&\text{total: } (41.047,\ 5.060), \qquad \text{distance} = \sqrt{41.047^2 + 5.060^2} \approx 41.36 \text{ km}\\
&\text{bearing from port} = \operatorname{atan2}(\text{east}, \text{north}) = \operatorname{atan2}(41.047, 5.060) \approx 82.97^\circ.
\end{aligned}$$

This matches the law-of-cosines answer and also gives the direction. Swapping the arguments to atan2 converts between the math convention (anticlockwise from east) and the compass convention (clockwise from north). Swapping them **by accident** is a classic bug.

Computer-science connection: angles in games and graphics

- **Facing and steering:** `angle = atan2(dy, dx)` to aim. A heading becomes a velocity with `vx = speed*cos(angle)`, `vy = speed*sin(angle)`.
- **Screen coordinates:** on most screens y points **down**, so a positive angle rotates **clockwise** on screen. Many "my rotation goes the wrong way" bugs come from this.
- **Angle wrapping:** to turn the short way from heading `a` to heading `b`, compute `d = (b - a + pi) % (2*pi) - pi` (in Python, where `%` returns a non-negative result for a positive divisor), which gives the signed difference in $[-\pi, \pi)$. Naive subtraction makes a turret spin 350° the wrong way.
- **Avoid trig when you can:** the angle between two vectors comes from the dot product (Volume 2, Chapter 26), rotations compose as matrices (Volume 2, Chapter 27), and 3D engines use quaternions to avoid "gimbal lock". Angles are for humans and input. Vectors and matrices are for the engine.
- **Field of view:** a camera with horizontal field of view ϕ at distance d sees a width of $2d\tan(\phi/2)$.

13.9.3. Degrees versus radians in math libraries

```python
import math

print(math.sin(90))                 # radians! not 1
print(math.sin(math.radians(90)))   # convert first
print(math.cos(math.pi / 2))        # not exactly 0
print(math.degrees(math.atan2(1, 1)))
```

Try it: See the degrees bug.

```
0.8939966636005579
1.0
6.123233995736766e-17
45.0
```



Classic trig bugs in code

- **Units:** `sin(45)` where `sin(radians(45))` was meant. Libraries take and return radians, while UIs and config files usually use degrees. Name variables with their units (`heading_deg`, `theta_rad`) and convert at the boundary.
- **atan vs atan2:** using `atan(y/x)` loses the quadrant and can divide by zero.
- **Argument order:** it's `atan2(y, x)` in C, Python, Java, JavaScript and NumPy, but Excel's `ATAN2(x, y)` takes them the other way round.
- **Domain errors:** `acos(1.0000000002)`, typically from a normalised dot product that rounding has pushed just past 1, raises `ValueError` or returns `NaN`. Clamp to $[-1, 1]$ first.
- **Exact zeros:** `cos(pi/2)` is about 6×10^{-17}, not 0. Use tolerances.
- **Huge arguments:** `sin(1e22)` is computed correctly by good libraries, but it's meaningless if `1e22` was itself produced with rounding error. Keep angles reduced (wrap them) as you go.

13.9.4. Synthesising sound with sine waves

A digital audio signal is a list of **samples**, numbers taken at a fixed **sample rate** (44,100 per second for CD audio). A pure tone of frequency f Hz and amplitude A is

$$s[n] = A \sin\left(2\pi f \frac{n}{f_s}\right), \qquad n = 0, 1, 2, \ldots,$$

where f_s is the sample rate and n/f_s is the time of sample n in seconds. Here is a complete, standard-library-only synthesiser that writes a one-second chord (A3, its octave, and a fifth above that) to a WAV file:

```python
import math, struct, wave

RATE = 44_100                       # samples per second
partials = [(220, 0.5), (440, 0.25), (660, 0.125)]   # (Hz, amplitude)

frames = bytearray()
for n in range(RATE):               # one second
    t = n / RATE
    s = sum(a * math.sin(2 * math.pi * f * t) for f, a in partials)
    frames += struct.pack("<h", int(32767 * s))       # 16-bit signed sample

with wave.open("chord.wav", "wb") as w:
    w.setnchannels(1)
    w.setsampwidth(2)
    w.setframerate(RATE)
    w.writeframes(bytes(frames))
print(len(frames) // 2, "samples written")
```

```
44100 samples written
```

The amplitudes add up to 0.875, so the sum never goes past full scale and doesn't **clip**.

13.9.5. A Fourier teaser: finding the sines inside a signal

Fourier analysis runs synthesis in reverse: given the summed waveform, recover which frequencies are in it and how strong each one is. NumPy's FFT does this in $O(n \log n)$ time. (That's the logarithm from Chapter 12 again.) *Try it:* See the spectrum live ; Find the three sines .

PYTHON

```python
import numpy as np

RATE = 8000
t = np.arange(RATE) / RATE                      # one second of samples
signal = (0.5 * np.sin(2 * np.pi * 220 * t)
          + 0.25 * np.sin(2 * np.pi * 440 * t)
          + 0.125 * np.sin(2 * np.pi * 660 * t))

spectrum = np.abs(np.fft.rfft(signal)) * 2 / len(signal)   # amplitude per frequency
freqs = np.fft.rfftfreq(len(signal), d=1 / RATE)
for i in np.argsort(spectrum)[-3:][::-1]:
    print(f"{freqs[i]:6.1f} Hz  amplitude {spectrum[i]:.3f}")
```

Try it: Run this in your browser .

```
 220.0 Hz  amplitude 0.500
 440.0 Hz  amplitude 0.250
 660.0 Hz  amplitude 0.125
```

The FFT recovers exactly the three sines that went in. In Chapter 15 you'll see why it uses complex numbers: Euler's formula $e^{i\theta} = \cos\theta + i\sin\theta$ packs a sine and a cosine into one exponential, which ties this chapter to the previous one.

Computer-science connection: where sines live in software

- **Audio:** oscillators, filters, equalisers, pitch detection, MP3/AAC (a modified discrete cosine transform).
- **Images and video:** JPEG and most video codecs use the discrete cosine transform on 8×8 blocks.
- **Communications:** Wi-Fi, 4G/5G and DSL use OFDM, which sends data on many sine carriers at once and decodes them with an FFT.
- **Animation:** `y = base + A*sin(2*pi*t/period)` gives smooth bobbing and pulsing. Ease-in-out curves are often half a cosine wave.
- **Machine learning:** Transformer models encode token positions with sines and cosines of geometrically spaced frequencies ("sinusoidal positional encoding"), and rotary position embeddings (RoPE) apply literal 2D rotations to pairs of coordinates.
- **Geography:** the haversine formula for great-circle distance between two latitude/longitude points is spherical trigonometry.



Build a synth

Sliders for the amplitudes of the first eight harmonics of a 220 Hz tone. Show the summed waveform and **play** it live. Presets: odd harmonics with amplitude $1/k$ (approaching a square wave), all harmonics with $1/k$ (a sawtooth). Then show the FFT bar chart of the result so synthesis and analysis appear side by side.

Try it: Harmonics towards a square wave · All harmonics make a sawtooth · Odd harmonics make a square



Rotate and aim

A turret at the centre of the screen tracks the mouse. Toggle between `atan(dy/dx)` and `atan2(dy, dx)` and watch the naive version flip the wrong way in two quadrants. Add a toggle for "y points down" to see the clockwise/anticlockwise flip, and a "degrees bug" toggle that passes degrees to `sin` / `cos`.

Try it: Rotate and aim

13.10. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Unit circle and waves
- Graphing calculator

- Geometry: triangles, circles, polygons, solids
- Spectrum analyser
- Fourier series
- Language sandbox

13.11. Exercises

Recognition

1. In a right triangle, the side opposite θ is 5 and the hypotenuse is 13. Find $\sin\theta$, $\cos\theta$, $\tan\theta$.
2. Convert to radians: 45°, 210°, −90°. Convert to degrees: $\frac{2\pi}{3}$, $\frac{7\pi}{4}$, 1.
3. State the sign of: $\sin 200^\circ$, $\cos 100^\circ$, $\tan 300^\circ$, $\cos(-50^\circ)$.
4. Give amplitude, period, phase shift and midline of $y = 4\cos(3x - \pi) - 1$.

Application

5. A kite string is 60 m long and makes 52° with the ground. How high is the kite (ignore sag)?
6. Find exact values: $\sin 300^\circ$, $\cos\frac{3\pi}{4}$, $\tan 225^\circ$, $\cos 15^\circ$.
7. Solve $2\cos x + 1 = 0$ for $0 \le x < 2\pi$.
8. $\cos\theta = -\frac{5}{13}$, θ in quadrant III. Find $\sin\theta$, $\tan\theta$ and $\sin 2\theta$.
9. Triangle with $a = 8$, $b = 11$, $C = 38^\circ$. Find c.
10. Convert $(-1, -\sqrt{3})$ to polar coordinates with $\theta \in (-\pi, \pi]$. Convert $(r, \theta) = (6, -\frac{\pi}{3})$ to Cartesian.

Problem solving

11. From the top of a 40 m lighthouse, the angle of depression to a boat is 7°. How far is the boat from the base of the lighthouse?
12. Rotate the triangle with vertices $(1, 0)$, $(3, 0)$, $(3, 2)$ by 45° anticlockwise about $(1, 0)$. Give exact coordinates.
13. Prove the identity $\dfrac{\sin 2x}{1 + \cos 2x} = \tan x$.
14. A Ferris wheel of radius 20 m has its centre 22 m above the ground and turns once every 4 minutes. A rider boards at the bottom at $t = 0$. Write their height $h(t)$ (metres, t in minutes) and find the first time they're 32 m up.
15. A function `heading_to(dx, dy)` must return a compass bearing in [0°, 360°) (clockwise from north) for a screen where y points **down** and north is up the screen. Write it.
16. What length does a pendulum need for a period of exactly 1 s? ($g = 9.81\ \text{m/s}^2$.)

▼ **Answers**

1. Adjacent $= \sqrt{13^2 - 5^2} = 12$. $\sin\theta = \frac{5}{13}$, $\cos\theta = \frac{12}{13}$, $\tan\theta = \frac{5}{12}$.
2. $\frac{\pi}{4}$, $\frac{7\pi}{6}$, $-\frac{\pi}{2}$; 120°, 315°, $180/\pi \approx 57.30°$.
3. 200° is in quadrant III, so sin is negative. 100° is in II, so cos is negative. 300° is in IV, so tan is negative. Cosine is even, so $\cos(-50°) = \cos 50°$ is positive.
4. $4\cos\left(3\left(x - \frac{\pi}{3}\right)\right) - 1$: amplitude 4, period $\frac{2\pi}{3}$, shift $\frac{\pi}{3}$ to the right, midline $y = -1$.
5. $h = 60\sin 52° = 60 \times 0.788011 \approx 47.3$ m.
6. $\sin 300° = -\sin 60° = -\frac{\sqrt{3}}{2}$; $\cos 135° = -\frac{\sqrt{2}}{2}$; $\tan 225° = +\tan 45° = 1$; $\cos(45° - 30°) = \frac{\sqrt{2}}{2}\cdot\frac{\sqrt{3}}{2} + \frac{\sqrt{2}}{2}\cdot\frac{1}{2} = \frac{\sqrt{6}+\sqrt{2}}{4} \approx 0.9659$.
7. $\cos x = -\frac{1}{2}$. Reference angle $\frac{\pi}{3}$, cosine negative in II and III: $x = \pi - \frac{\pi}{3} = \frac{2\pi}{3}$ and $x = \pi + \frac{\pi}{3} = \frac{4\pi}{3}$.
8. $\sin^2\theta = 1 - \frac{25}{169} = \frac{144}{169}$, and sine is negative in III, so $\sin\theta = -\frac{12}{13}$. $\tan\theta = \frac{-12/13}{-5/13} = \frac{12}{5}$. $\sin 2\theta = 2\sin\theta\cos\theta = 2(-\frac{12}{13})(-\frac{5}{13}) = \frac{120}{169}$.
9. $c^2 = 64 + 121 - 176\cos 38° = 185 - 176 \times 0.788011 = 185 - 138.69 = 46.31$, so $c \approx 6.81$.
10. $r = \sqrt{1+3} = 2$. The point is in quadrant III: $\theta = \operatorname{atan2}(-\sqrt{3}, -1) = -\frac{2\pi}{3}$. Second part: $x = 6\cos(-\frac{\pi}{3}) = 3$, $y = 6\sin(-\frac{\pi}{3}) = -3\sqrt{3} \approx -5.196$.
11. The angle of depression from the top equals the angle of elevation from the boat (alternate angles), so $\tan 7° = 40/d$ and $d = 40/\tan 7° = 40/0.122785 \approx 325.8$ m.
12. Translate by $(-1, 0)$: the vertices become $(0,0), (2,0), (2,2)$. Rotate by 45° using $\cos = \sin = \frac{\sqrt{2}}{2}$: $(0,0)$; $(2,0) \mapsto (\sqrt{2}, \sqrt{2})$; $(2,2) \mapsto (2\frac{\sqrt{2}}{2} - 2\frac{\sqrt{2}}{2},\ 2\frac{\sqrt{2}}{2} + 2\frac{\sqrt{2}}{2}) = (0, 2\sqrt{2})$. Translate back: $(1,0)$, $(1+\sqrt{2}, \sqrt{2})$, $(1, 2\sqrt{2})$.
13. Left side: $\dfrac{2\sin x\cos x}{1 + (2\cos^2 x - 1)} = \dfrac{2\sin x\cos x}{2\cos^2 x} = \dfrac{\sin x}{\cos x} = \tan x$ (where defined).
14. $h(t) = 22 - 20\cos\left(\frac{2\pi}{4}t\right) = 22 - 20\cos\left(\frac{\pi t}{2}\right)$. Check: $h(0) = 2$, the bottom. For 32 m: $\cos(\pi t/2) = -\frac{1}{2}$, so $\pi t/2 = \frac{2\pi}{3}$ and $t = \frac{4}{3}$ minutes (80 s).
15. North is up the screen, which is $-y$ in screen coordinates, and east is $+x$. Bearing $= \operatorname{atan2}(\text{east}, \text{north}) = \operatorname{atan2}(dx, -dy)$. In Python: `return math.degrees(math.atan2(dx, -dy)) % 360`.
16. $T = 2\pi\sqrt{L/g} \Rightarrow L = g\left(\frac{T}{2\pi}\right)^2 = 9.81/(4\pi^2) \approx 0.2485$ m.

13.12. Mastery check

1. A tree casts a 25 m shadow when the Sun is 40° above the horizon. How tall is it?
2. Without a calculator: $\sin\frac{7\pi}{6}$, $\cos\left(-\frac{\pi}{4}\right)$, $\tan\frac{2\pi}{3}$.
3. Explain in one or two sentences why $\sin^2\theta + \cos^2\theta = 1$ holds for every angle.
4. Use the angle-sum formula to find the exact value of $\cos 105°$.
5. Triangle with sides 7, 8, 5: find the largest angle.
6. Rotate $(4, -2)$ by 90° clockwise about the origin.
7. Why does `math.atan2(-1, -1)` give a different answer from `math.atan(-1 / -1)`? Give both answers in degrees.
8. Write a sinusoid with amplitude 3, period 0.01 s, midline 0, starting at 0 and rising at $t = 0$. What is its frequency?

Decision rule: All correct without hesitation → move on to Chapter 14 or Chapter 15. Stuck on 1–2 → reread Section 13.1 and Section 13.3. Stuck on 3–4 → reread Section 13.5. Stuck on 5 → reread Section 13.6. Stuck on 6–7 → reread Section 13.9. Stuck on more → work the section exercises first.

▼ **Mastery check answers**

1. $h = 25\tan 40° = 25 \times 0.839100 \approx 20.98$ m.
2. $\frac{7\pi}{6} = 210°$, quadrant III: $-\frac{1}{2}$. Cosine is even: $\frac{\sqrt{2}}{2}$. $120°$, quadrant II: $-\tan 60° = -\sqrt{3}$.
3. $(\cos\theta, \sin\theta)$ is by definition a point on the unit circle $x^2 + y^2 = 1$, whatever the angle.
4. $\cos(60° + 45°) = \frac{1}{2}\cdot\frac{\sqrt{2}}{2} - \frac{\sqrt{3}}{2}\cdot\frac{\sqrt{2}}{2} = \frac{\sqrt{2}-\sqrt{6}}{4} \approx -0.2588$.
5. The largest angle is opposite the side of length 8: $\cos\theta = \dfrac{7^2 + 5^2 - 8^2}{2\cdot 7\cdot 5} = \dfrac{10}{70} = \frac{1}{7}$, so $\theta = \cos^{-1}(1/7) \approx 81.79°$.
6. Clockwise 90° is $\theta = -90°$: $x' = x\cos(-90°) - y\sin(-90°) = 0 - (-2)(-1) = -2$, $y' = x\sin(-90°) + y\cos(-90°) = -4 + 0 = -4$. The answer is $(-2, -4)$. In general a clockwise quarter-turn sends $(x, y) \mapsto (y, -x)$.
7. $-1/-1 = 1$, so `atan` sees only the ratio and returns 45°. `atan2` sees both negative signs, places the point in quadrant III, and returns −135°.
8. Frequency $f = 1/0.01 = 100$ Hz, $\omega = 2\pi f = 200\pi$, so $y = 3\sin(200\pi t)$.

Chapter 14. Sequences, Series, and Summation Notation

Chapter map: a sequence is an ordered list of numbers produced by a rule; a series is what you get when you add the terms up. This chapter teaches you to describe sequences two ways (explicitly and recursively), to read and write the compact notations Σ and Π, to sum arithmetic and geometric series in closed form, and to ask the deep question "does an infinite sum have a value?" Along the way you will meet the same mathematics hiding in nested loops, growable arrays, recursive algorithms, and memoisation.

Why this chapter matters: a mortgage payment, a pension projection, the level of a medicine in your blood after a week of daily doses, the number of matches in a league season and the running time of a nested loop are all sums of sequences. With two formulas (arithmetic and geometric) you can compute them in one line instead of a spreadsheet of a thousand rows, and you can tell in advance whether a process settles down or runs away.

Where this chapter sits

You will need: arithmetic with fractions and negatives (Chapter 2), exponents and their laws (Chapter 3), manipulating expressions and solving linear equations (Chapter 4, Chapter 5), the idea of a function (Chapter 11), and exponential growth and logarithms (Chapter 12).

This chapter unlocks: limits of sequences and the idea of convergence (Volume 2, Chapter 16), Riemann sums and the definite integral (Volume 2, Chapter 19), Taylor series and power series (glimpsed in Chapter 15 and Volume 2, Chapter 18), proof by induction (Volume 3, Chapter 22), counting arguments (Volume 3, Chapter 24), expected values (Volume 3, Chapter 29), and the running-time analysis you see across computer science.

14.1. Sequences: lists with a rule

14.1.1. The problem

You put $1{,}000 into an account paying 4% interest per year. What is the balance after 1 year, 2 years, 10 years? A ball is dropped from 10 m and each bounce reaches 60% of the previous height. How high is the seventh bounce? A theatre has 20 seats in the front row and each row behind has 2 more. How many seats are in row 30?

Each question asks for one item in an ordered list of numbers that is generated by a rule. That list is a **sequence**, and the questions become: *given the rule, find the nth item* — and, later, *add the items up*.

14.1.2. Intuition

Think of a sequence as an array that never ends, indexed by position: `a[1], a[2], a[3], …`. The interesting thing is never the individual numbers; it is the rule that produces them. Once you know the rule, you can answer questions about item 1,000,000 without writing out the first 999,999.

There are two natural ways to state a rule, and programmers already know both:

- **Explicit (closed-form):** a formula that gives the nth term directly from n. Like a pure function `a(n)`.
- **Recursive:** a starting value plus a rule for getting each term from the previous one(s). Like a loop that updates a variable, or a recursive function.



Sequence, term, index

A **sequence** is an ordered list of numbers $a_1, a_2, a_3, \ldots$. Each number is a **term**. The subscript is the **index** (position): a_n is read "a sub n" and means "the term in position n". The whole sequence is written (a_n) or $\{a_n\}_{n\geq 1}$. A sequence may be **finite** (it stops) or **infinite** (it goes on forever). Formally, an infinite sequence is just a function whose domain is the positive integers: a_n is another way of writing $a(n)$.



Explicit formula and recursive definition

An **explicit formula** (or **closed form**, or **general term**) gives a_n directly in terms of n, for example $a_n = n^2$.

A **recursive definition** (or **recurrence relation**) gives one or more **initial conditions** and a rule expressing a_n in terms of earlier terms, for example $a_1 = 1,\ a_n = a_{n-1} + 2n - 1$. Without the initial condition a recurrence does not pin down a single sequence.

14.1.3. Why both descriptions matter

Here is the same sequence, the perfect squares, described both ways:

$$\textbf{explicit: } a_n = n^2 \qquad\qquad \textbf{recursive: } a_1 = 1, \quad a_n = a_{n-1} + (2n-1) \text{ for } n \geq 2.$$

Check the recursive version: $a_2 = 1 + 3 = 4$, $a_3 = 4 + 5 = 9$, $a_4 = 9 + 7 = 16$. Each square is the previous square plus the next odd number. The explicit form is better for jumping straight to $a_{1000} = 1{,}000{,}000$; the recursive form often reflects *how the quantity actually arises* — the bank adds interest to last year's balance, the ball's bounce depends on the previous bounce. Much of this chapter is about converting a recursive description (which is easy to write down from a story) into an explicit one (which is easy to compute and reason about).

```python
def squares_explicit(n):
    return n * n

def squares_recursive(n):
    a = 1                       # initial condition a_1 = 1
    for k in range(2, n + 1):   # apply a_k = a_{k-1} + (2k - 1)
        a = a + (2 * k - 1)
    return a

print([squares_explicit(n) for n in range(1, 8)])
print([squares_recursive(n) for n in range(1, 8)])
```

```
[1, 4, 9, 16, 25, 36, 49]
[1, 4, 9, 16, 25, 36, 49]
```

14.1.4. Worked example: finding a rule from terms

Find an explicit formula for $2, 5, 10, 17, 26, \ldots$.

Step 1 — look at differences. $5-2=3,\ 10-5=5,\ 17-10=7,\ 26-17=9$. The differences are consecutive odd numbers, which is the signature of squares (as we just saw).

Step 2 — compare with n^2. For $n=1,2,3,4,5$: $n^2 = 1, 4, 9, 16, 25$. Each given term is exactly one more.

Step 3 — state and check. $a_n = n^2 + 1$. Check $n=4$: $16+1=17$. ✓



A few terms never determine a sequence

"What comes next: 1, 2, 4, 8, 16, ...?" Most people say 32, and the rule $a_n = 2^{n-1}$ agrees. But the number of regions formed by joining n points on a circle in general position goes 1, 2, 4, 8, 16, **31**. Finitely many terms are consistent with infinitely many rules. In mathematics a sequence is defined by its rule, not by its first few terms; "spot the pattern" puzzles are guesses, not deductions.



Indexing from 0 or 1

Mathematicians often start at a_1; programmers (and many mathematicians when it is more natural) start at a_0. Neither is wrong, but mixing them causes off-by-one errors exactly as in code. Always check which index the first term has before using a formula. In this chapter we say explicitly which we are using.

14.2. Arithmetic and geometric sequences

Two families of sequences account for a remarkable fraction of real applications: those that grow by *adding* the same amount each step and those that grow by *multiplying* by the same amount each step.

14.2.1. The problem

The theatre (20 seats in row 1, two more each row) and the savings account (multiply by 1.04 each year) are the two prototypes. We want explicit formulas for both.



Arithmetic sequence

An **arithmetic sequence** (or **arithmetic progression**, AP) adds a fixed **common difference** d each step: $a_n = a_{n-1} + d$. Its explicit form, with first term a_1, is $a_n = a_1 + (n-1)d$. The word is pronounced a-rith-MET-ic (stress on the third syllable) when used as an adjective.



Geometric sequence

A **geometric sequence** (or **geometric progression**, GP) multiplies by a fixed **common ratio** r each step: $a_n = r\,a_{n-1}$. Its explicit form, with first term a_1, is $a_n = a_1 r^{\,n-1}$.

14.2.2. Why the formulas work

For an arithmetic sequence, getting from term 1 to term n takes $n-1$ steps, each adding d:

$$\begin{aligned} a_2 &= a_1 + d \\ a_3 &= a_2 + d = a_1 + 2d \\ a_4 &= a_3 + d = a_1 + 3d \\ &\vdots \\ a_n &= a_1 + (n-1)d. \end{aligned}$$

The "$n-1$" is the classic fence-post count: n posts, $n-1$ gaps. For a geometric sequence the same unrolling multiplies instead of adds, giving $a_n = a_1 r^{n-1}$.

Seen as functions of n, an arithmetic sequence is a **linear function** sampled at whole numbers (slope d), and a geometric sequence is an **exponential function** sampled at whole numbers (base r). The picture makes the difference unforgettable.

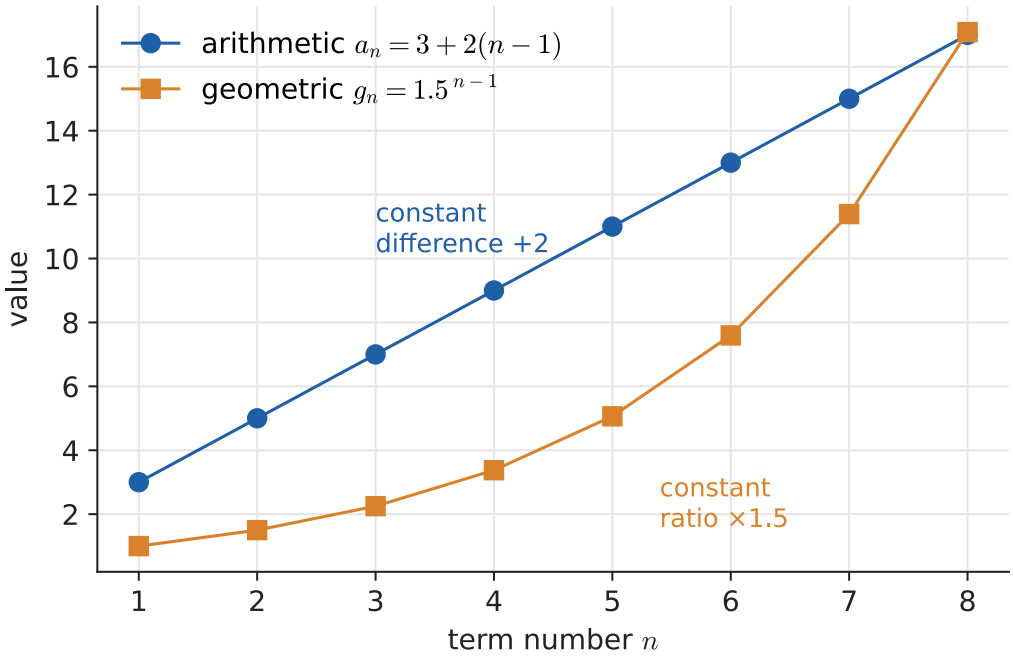


Figure 115. Arithmetic sequences sit on a straight line; geometric sequences sit on an exponential curve.

14.2.3. The method

Example 75. Procedure — identify and describe an arithmetic or geometric sequence

Input: several consecutive terms of a sequence.

Steps:

1. Compute consecutive differences $a_{k+1} - a_k$. If they are all equal to some d, the sequence is arithmetic: $a_n = a_1 + (n-1)d$.
2. Otherwise compute consecutive ratios a_{k+1}/a_k (all terms nonzero). If they are all equal to some r, the sequence is geometric: $a_n = a_1 r^{n-1}$.
3. If neither, try second differences (constant second differences mean a quadratic in n) or compare with known sequences.
4. Check the formula against a term you did not use to derive it.

Output: an explicit formula a_n.

14.2.4. Worked examples

Theatre seats. $a_1 = 20$, $d = 2$. Row 30 has

$$a_{30} = 20 + (30-1)\cdot 2 = 20 + 58 = 78 \text{ seats.}$$

Which term? In $7, 11, 15, 19, \ldots$, is 203 a term, and if so which one? Here $d = 4$, so $a_n = 7 + 4(n-1) = 4n + 3$. Solve $4n + 3 = 203$: $4n = 200$, $n = 50$. Yes, 203 is the 50th term. (If solving had given a non-integer n, the number would not be in the sequence.)

Bouncing ball. The first bounce reaches 6 m (60% of 10 m), so $a_1 = 6$, $r = 0.6$. The seventh bounce reaches

$$a_7 = 6 \cdot 0.6^6 = 6 \cdot 0.046656 = 0.279936 \approx 0.28 \text{ m}.$$

How many years to double? At 4% per year, the balance after n years is $1000 \cdot 1.04^n$ (indexing from $n = 0$, the initial deposit). Doubling means $1.04^n = 2$, so, taking logarithms (Chapter 12),

$$n = \frac{\ln 2}{\ln 1.04} = \frac{0.693147}{0.039221} \approx 17.67,$$

so the balance first exceeds \$2,000 after 18 years.



Negative and fractional ratios

A geometric sequence can have $r < 0$: $3, -6, 12, -24, \ldots$ has $r = -2$ and alternates in sign. With $0 < |r| < 1$ the terms shrink towards zero (decay). With $r = 1$ the sequence is constant; with $r = 0$ it collapses to zero after the first term. Check the sign of r carefully: $(-2)^{n-1}$ and -2^{n-1} are different things.

14.2.5. Where you'd use this

- **Everyday (arithmetic):** a taxi charges \$3.50 to start plus \$2.20 per mile. The fare after n miles is the arithmetic sequence $a_n = 3.50 + 2.20n$, so a 12-mile ride costs $3.50 + 26.40 = \$29.90$. Any "fixed fee plus a rate" price works this way: phone plans, gym memberships, parking.
- **Music (geometric):** on a piano each semitone raises the frequency by the same **factor**, $2^{1/12} \approx 1.0595$, so twelve semitones double it (an octave). The notes form a geometric sequence. From A at 440 Hz, the C three semitones up is $440 \times 2^{3/12} \approx 523.25$ Hz.
- **Science (geometric):** carbon-14 halves every 5,730 years, so after n half-lives a fraction 0.5^n remains. A bone with 12.5% of its original carbon-14 has been through 3 half-lives: it is about 17,000 years old.
- **Computing (geometric):** when a network request fails, well-behaved clients retry with **exponential backoff**: wait 0.1 s, then 0.2 s, 0.4 s, and so on ($a_k = 0.1 \times 2^{k-1}$). The 8th retry waits $0.1 \times 2^7 = 12.8$ s. Doubling the gap stops thousands of clients from hammering a struggling server in lock-step.



Arithmetic and geometric means

The middle term of three consecutive terms of an arithmetic sequence is the average of its neighbours, $\frac{a+c}{2}$ — hence the **arithmetic mean**. In a geometric sequence the middle term is $\sqrt{ac}$, the **geometric mean**. That is where the names come from. The geometric mean is the right average for growth rates: if an investment grows 50% one year and falls 50% the next, the average growth factor is $\sqrt{1.5 \times 0.5} \approx 0.866$, not 1.

US curriculum

Arithmetic and geometric sequences appear in **Algebra I** (grades 8–9), where they are treated as linear and exponential functions with domain the positive integers (Common Core HSF-BF.A.2, HSF-LE.A.2). Recursive notation is introduced there too. They are revisited with series in **Algebra II** and **Precalculus**.

14.3. Summation notation: Σ and Π

14.3.1. The problem

Before adding up sequences we need a way to *write* a sum of many terms without ellipses and ambiguity. "$1 + 4 + 9 + \cdots + 100$" relies on the reader guessing the pattern. Mathematics has a precise, compact notation for this — and it is exactly a `for` loop.

14.3.2. Reading Σ piece by piece

Here is the notation in full:

$$\sum_{i=1}^{n} x_i$$

Read it as "the sum, for i from 1 to n, of x sub i". Each piece has a job:

- Σ — the capital Greek letter sigma, for "S" as in **sum**. It says: add things up.
- i (below the sigma, before the equals sign) — the **index of summation**, a counter variable. It is a **dummy variable**: it exists only inside the sum, just as a loop variable exists only inside the loop. Writing k or j instead changes nothing.
- 1 (below, after the equals sign) — the **lower bound**: the first value the index takes.
- n (above) — the **upper bound**: the last value the index takes. Both bounds are **inclusive**.
- x_i (to the right) — the **summand**, the expression evaluated for each value of the index and then added.

So $\sum_{i=1}^{n} x_i$ means $x_1 + x_2 + \cdots + x_n$. The index steps by 1 from lower bound to upper bound; there is no "step" parameter.

```python
# Sigma notation is a for-loop
def sigma(lo, hi, term):
    total = 0
    for i in range(lo, hi + 1):   # inclusive upper bound, like the maths
        total += term(i)
    return total

print(sigma(1, 100, lambda i: i))          # 1 + 2 + ... + 100
print(sigma(1, 5, lambda k: k * k))        # 1 + 4 + 9 + 16 + 25
print(sum(k * k for k in range(1, 6)))     # the idiomatic one-liner
```

```
5050
55
55
```

Note the `hi + 1`: Python's `range` excludes its end point, sigma includes it. That single fact causes most off-by-one errors when translating formulas into code.



Sigma notation

$\sum_{i=m}^{n} f(i) = f(m) + f(m+1) + \cdots + f(n)$. The **index** i is a dummy variable; m and n are the **lower** and **upper bounds** (inclusive); $f(i)$ is the **summand**. If $n < m$ the sum is **empty** and equals 0. The number of terms is $n - m + 1$.

14.3.3. Worked examples: reading and writing sigma

Expand. $\sum_{k=2}^{5}(2k-1)$: the index runs over $k = 2, 3, 4, 5$, giving $3 + 5 + 7 + 9 = 24$.

Expand with an index that appears as an exponent.
$\sum_{j=0}^{4} 3^j = 3^0 + 3^1 + 3^2 + 3^3 + 3^4 = 1 + 3 + 9 + 27 + 81 = 121$.

Summand not depending on the index. $\sum_{i=1}^{6} 5 = 5 + 5 + 5 + 5 + 5 + 5 = 30$. The summand is evaluated six times, even though it ignores i.

Compress. Write $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{1024}$ in sigma notation. The terms are $\frac{1}{2^k}$ and $1024 = 2^{10}$, so the sum is $\sum_{k=1}^{10} \frac{1}{2^k}$.

Compress an alternating sum. $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5}$. The sign flips each term, which $(-1)^{k+1}$ does neatly (positive when k is odd): $\sum_{k=1}^{5} \frac{(-1)^{k+1}}{k}$.

14.3.4. Product notation Π

The same idea with multiplication uses capital pi, $\prod$, for "P" as in **product**:

$$\prod_{k=1}^{n} a_k = a_1 \cdot a_2 \cdots a_n.$$

The most famous example is the factorial: $n! = \prod_{k=1}^{n} k$. Another, from probability (Volume 3, Chapter 29): the chance that n people all have different birthdays (ignoring leap years) is

$$\prod_{k=0}^{n-1} \frac{365 - k}{365} = \frac{365}{365} \cdot \frac{364}{365} \cdots \frac{365 - n + 1}{365},$$

which first falls below one-half at $n = 23$ — the "birthday paradox".

Product notation and the empty product

$\prod_{i=m}^{n} f(i) = f(m) \cdot f(m+1) \cdots f(n)$. An **empty product** (upper bound below lower bound) equals 1, the identity for multiplication, just as an empty sum equals 0, the identity for addition. This is why $\mathbf{0! = 1}$. In code: `total = 0` before a summing loop, `prod = 1` before a multiplying loop.

14.3.5. Manipulating sums

Because a sum is just repeated addition, the ordinary laws of arithmetic give a handful of rules. Each one is the distributive, commutative, or associative law in disguise.

$$\sum_{i=1}^{n} c\,a_i = c\sum_{i=1}^{n} a_i \qquad \textbf{(pull out a constant factor: distributive law)}$$

$$\sum_{i=1}^{n}(a_i + b_i) = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i \qquad \textbf{(split a sum: commutative/associative laws)}$$

$$\sum_{i=1}^{n} c = nc \qquad \textbf{(n copies of a constant)}$$

$$\sum_{i=1}^{n} a_i = \sum_{i=1}^{m} a_i + \sum_{i=m+1}^{n} a_i \qquad \textbf{(split the range, } 1 \le m < n\textbf{)}$$

$$\sum_{i=1}^{n} a_i = \sum_{j=0}^{n-1} a_{j+1} \qquad \textbf{(shift the index: } j = i - 1\textbf{)}$$

The first two together say that sigma is **linear** — the same property you will meet again for derivatives (Volume 2, Chapter 17), integrals (Volume 2, Chapter 19), expected values (Volume 3, Chapter 29), and matrices (Volume 2, Chapter 27).

What you cannot do

Sums do **not** distribute over products or powers:

$$\sum_i a_i b_i \neq \Big(\sum_i a_i\Big)\Big(\sum_i b_i\Big), \qquad \sum_i a_i^2 \neq \Big(\sum_i a_i\Big)^2.$$

Try $a = (1, 2)$: $\sum a_i^2 = 1 + 4 = 5$ but $(\sum a_i)^2 = 3^2 = 9$. Also, you can pull out only factors that do not depend on the index: $\sum_i i\,a_i \neq i \sum_i a_i$ (the right-hand side is not even meaningful, since i outside the sum is undefined).

Worked example. Evaluate $\sum_{i=1}^{10}(3i + 2)$ using the rules and the fact (proved in the next section) that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

$$\begin{aligned}\sum_{i=1}^{10}(3i+2) &= \sum_{i=1}^{10}3i + \sum_{i=1}^{10}2 && \text{(split the sum)}\\ &= 3\sum_{i=1}^{10} i + 10\cdot 2 && \text{(pull out 3; ten copies of 2)}\\ &= 3\cdot\frac{10\cdot 11}{2} + 20 && \text{(Gauss's formula)}\\ &= 165 + 20 = 185.\end{aligned}$$

Check by brute force: the terms are $5, 8, 11, \dots, 32$; that is an arithmetic sequence of ten terms with first 5 and last 32, and (as we are about to see) its sum is $10\cdot\frac{5+32}{2} = 185$. ✓

14.3.6. Where you'd use this

Every average, total and score you have ever been given is a sigma expression.

- **School and work:** a course grade is a **weighted sum** $\sum_i w_i s_i$, with weights w_i adding to 1. Coursework worth 30% at 72 and an exam worth 70% at 64 gives $0.3\times 72 + 0.7\times 64 = 21.6 + 44.8 = 66.4$.
- **Science and statistics:** the mean of n measurements is $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, and the variance is $\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2$ (Volume 3, Chapter 31). In physics, the centre of mass of point masses is $\sum m_i x_i / \sum m_i$. A 70 kg adult at 0 m and a 30 kg child at 3 m on a seesaw balance at $(0+90)/100 = 0.9$ m from the adult.
- **Computing:** a neuron in a neural network computes $\sum_i w_i x_i + b$ and then applies a simple function to the result. Training minimises a loss such as the mean squared error $\frac{1}{n}\sum_i (y_i-\hat{y}_i)^2$. In a search engine, a document's relevance score is often a weighted sum of term scores. All of these are `sum(…)` loops, and the sigma rules above are how their gradients get derived (Volume 2, Chapter 17).

Computer-science connection: Σ is reduce

$\sum_{i=m}^{n} f(i)$ is `sum(f(i) for i in range(m, n+1))`; $\prod$ is `math.prod(…)`. More abstractly, both are a **fold** (`reduce`) with an associative operation and its identity element: `reduce(add, xs, 0)` and `reduce(mul, xs, 1)`. The sigma rules are what make these folds parallelisable: because addition is associative and commutative, you can split the range across machines, sum each piece, and add the partial results (the "split the range" rule). That is precisely what a MapReduce `sum` or a GPU parallel reduction does. Floating-point addition is *not* exactly associative, which is why a parallel sum can differ in the last few bits from a sequential one (Volume 2, Chapter 32).

US curriculum

Sigma notation is usually introduced in **Algebra II** or **Precalculus** alongside series, and used heavily in **AP Calculus** (Riemann sums) and **AP Statistics** (means and variances). Product notation is rarely taught in school and usually first met in college discrete mathematics.

14.4. Arithmetic series: Gauss's trick

14.4.1. The problem

A **series** is the sum of the terms of a sequence. The partial sum $S_n = a_1 + a_2 + \cdots + a_n = \sum_{k=1}^{n} a_k$ answers questions such as "how many seats in the whole theatre?" or "how many operations does this loop perform?" Adding the terms one at a time is fine for ten terms and hopeless for a million. We want a closed form.



Series and partial sum

A **series** is a sum of the terms of a sequence. The **nth partial sum** is $S_n = \sum_{k=1}^{n} a_k$. An **arithmetic series** sums an arithmetic sequence; a **geometric series** sums a geometric one. The partial sums $S_1, S_2, S_3, \ldots$ themselves form a new sequence — this observation is the key to infinite series.

14.4.2. Intuition: pair the ends

The story (possibly embellished) is that the young Carl Friedrich Gauss, told to add the numbers 1 to 100 as busywork, wrote down 5050 almost immediately. The trick: pair the first and last terms, the second and second-to-last, and so on.

$$\underbrace{1+100}_{101} \quad \underbrace{2+99}_{101} \quad \underbrace{3+98}_{101} \quad \cdots \quad \underbrace{50+51}_{101}$$

Fifty pairs, each summing to 101, total 5050.

14.4.3. Why it works: write it twice

Pairing is fiddly when n is odd (one number is left unpaired). A cleaner version writes the sum forwards and backwards and adds the two lines:

$$\begin{aligned}
S &= 1 + 2 + 3 + \cdots + (n-1) + n \\
S &= n + (n-1) + (n-2) + \cdots + 2 + 1 \\
\hline
2S &= (n+1) + (n+1) + (n+1) + \cdots + (n+1) + (n+1) && \text{(add column by column)} \\
2S &= n(n+1) && \text{(there are } n \text{ columns)} \\
S &= \frac{n(n+1)}{2}.
\end{aligned}$$

The picture is even more convincing: a staircase of blocks of heights $1, 2, \ldots, n$, plus a copy turned upside down, fills an $n \times (n+1)$ rectangle exactly.

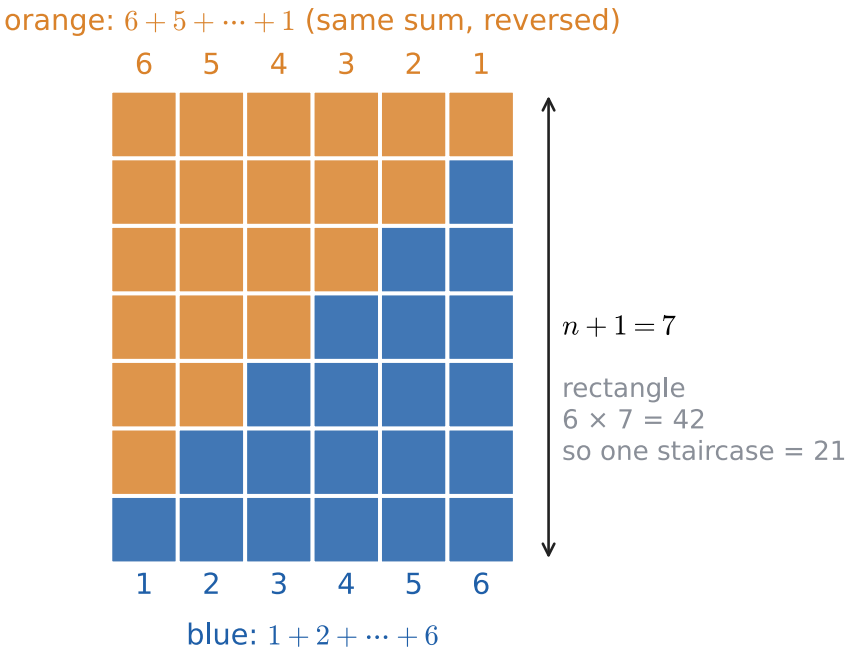


Figure 116. Two copies of the staircase 1 + 2 + ⋯ + 6 make a 6 × 7 rectangle, so one staircase has 21 blocks.

The same "write it twice" argument works for any arithmetic series, because in an arithmetic sequence the first plus the last equals the second plus the second-to-last, and so on (each step forward adds d to one and subtracts d from the other):

$$S_n = \sum_{k=1}^{n} \big(a_1 + (k-1)d\big) = \frac{n\,(a_1 + a_n)}{2} = \frac{n}{2}\big(2a_1 + (n-1)d\big).$$

In words: **number of terms times the average of the first and last term.** That phrasing is easy to remember and hard to misuse.



Triangular numbers

The numbers $T_n = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$ (1, 3, 6, 10, 15, …) are the **triangular numbers**, because that many dots can be arranged in a triangle. Note $T_n = \binom{n+1}{2}$, the number of ways to choose 2 items from $n+1$ (Volume 3, Chapter 24).

14.4.4. The method

Example 76. Procedure — sum an arithmetic series

Input: first term a_1, common difference d, and either the number of terms n or the last term a_n.

Steps:

1. If you know the last term but not n, find n from $a_n = a_1 + (n-1)d$, i.e. $n = \frac{a_n - a_1}{d} + 1$.
2. If you know n but not the last term, compute $a_n = a_1 + (n-1)d$.
3. Compute $S_n = \dfrac{n(a_1 + a_n)}{2}$.
4. Sanity-check: S_n should be n times something between the smallest and largest term.

Output: the sum S_n.

14.4.5. Worked examples

The whole theatre. 30 rows, $a_1 = 20$, $a_{30} = 78$ (found earlier):

$$S_{30} = \frac{30\,(20 + 78)}{2} = 15 \times 98 = 1470 \text{ seats.}$$

Odd numbers. The sum of the first n odd numbers $1 + 3 + 5 + \cdots + (2n-1)$: here $a_1 = 1$, $a_n = 2n - 1$, so

$$S_n = \frac{n\big(1 + (2n-1)\big)}{2} = \frac{n \cdot 2n}{2} = n^2,$$

confirming the "squares are sums of odd numbers" pattern from the start of the chapter.

Sum of multiples. Sum all multiples of 7 between 100 and 500. The first is $105 = 7 \cdot 15$, the last is $497 = 7 \cdot 71$. Number of terms: $71 - 15 + 1 = 57$. Sum: $\frac{57\,(105 + 497)}{2} = \frac{57 \times 602}{2} = 57 \times 301 = 17{,}157$.

14.4.6. Where you'd use this

- **A savings challenge:** save \$1 in week 1, \$2 in week 2, and so on up to \$52 in week 52. The total is $1 + 2 + \cdots + 52 = \frac{52 \times 53}{2} = \$1{,}378$. Doing it in reverse (start at \$52) gives the same total but front-loads the hard weeks.
- **Sport:** in a league where each of n teams plays every other team once, there are $0 + 1 + \cdots + (n-1) = \frac{n(n-1)}{2}$ fixtures. The Premier League's 20 teams play home and away, so a season has $2 \times \frac{20 \times 19}{2} = 380$ matches.
- **Physics:** a stone dropped from rest falls 4.9 m in the first second, 14.7 m in the next, then 24.5 m, 34.3 m, and so on. That's an arithmetic sequence with $d = 9.8$ m (that is, g). The total after n seconds is $\frac{n}{2}(4.9 + 4.9(2n-1)) = 4.9n^2$, which is the SUVAT formula $s = \frac{1}{2}gt^2$. Galileo noticed

the odd-number pattern 1 : 3 : 5 : 7 by rolling balls down ramps.

- **Computing:** the running time of every "compare each pair" algorithm is an arithmetic series, as Section 14.9 shows. A 10,000-row table checked pairwise for duplicates needs $\frac{10{,}000 \times 9{,}999}{2} \approx 5 \times 10^7$ comparisons. That is why databases sort or hash instead.

14.4.7. Alternative approach: sums of powers

Gauss's formula is the first of a family. Others you will occasionally need:

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}, \qquad \sum_{k=1}^{n} k^3 = \left(\frac{n(n+1)}{2}\right)^2.$$

For example $\sum_{k=1}^{100} k^2 = \frac{100 \cdot 101 \cdot 201}{6} = 338{,}350$. These are best *verified* by induction (Volume 3, Chapter 22) rather than derived here. For algorithm analysis the takeaway is the pattern: summing k^p up to n gives something that grows like $\frac{n^{p+1}}{p+1}$ — the discrete cousin of $\int x^p \, dx = \frac{x^{p+1}}{p+1}$ (Volume 2, Chapter 19).



Gauss's formula and the off-by-one

$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$ but $\sum_{k=0}^{n-1} k = \frac{(n-1)n}{2}$. A loop `for k in range(n)` runs $k = 0, \ldots, n-1$, so its sum of k is $\frac{n(n-1)}{2}$, not $\frac{n(n+1)}{2}$. When in doubt, test the formula on $n = 1$ or $n = 2$.

14.5. Geometric series: shift and subtract

14.5.1. The problem

You deposit $100 at the end of every year into an account paying 5% annually. How much is there after 10 years? The deposit made at the end of year 10 has earned nothing; the one from year 9 has earned one year's interest, and so on back to the first deposit, which has earned 9 years. The total is

$$100 + 100(1.05) + 100(1.05)^2 + \cdots + 100(1.05)^9 = \sum_{k=0}^{9} 100\,(1.05)^k,$$

a geometric series. Gauss's pairing trick does nothing here — the terms do not pair to a constant. We need a different idea.

14.5.2. Intuition and derivation: shift and subtract

Multiplying a geometric series by its ratio r *shifts every term one place along*. Subtracting the shifted copy from the original then cancels almost everything. Let

$$S = a + ar + ar^2 + \cdots + ar^{n-1}.$$

$$
\begin{aligned}
S &= a + ar + ar^2 + \cdots + ar^{n-1} && \\
rS &= \quad ar + ar^2 + \cdots + ar^{n-1} + ar^n && \text{(multiply every term by } r\text{)} \\
\hline
S - rS &= a - ar^n && \text{(all the middle terms cancel)} \\
S(1-r) &= a(1-r^n) && \text{(factor both sides)} \\
S &= \frac{a(1-r^n)}{1-r} && \text{(divide, valid when } r \neq 1\text{).}
\end{aligned}
$$

This is one of the most useful formulas in mathematics. In sigma notation:

$$\boxed{\sum_{k=0}^{n-1} ar^k = a\,\frac{1-r^n}{1-r} = a\,\frac{r^n-1}{r-1}, \qquad r \neq 1.}$$

The two forms are the same (multiply top and bottom by -1); use the second when $r > 1$ to keep things positive. If $r = 1$ every term is a and the sum is simply na.



Count carefully: the sum has n terms, from ar^0 to ar^{n-1}, and the exponent in the formula is the *number of terms*, n, which is one more than the largest exponent. A useful memory aid: "first term times (1 minus ratio to the number-of-terms) over (1 minus ratio)."

14.5.3. The method

Example 77. Procedure — sum a finite geometric series

Input: a sum of terms each a constant multiple of the previous one.

Steps:

1. Identify the first term a, the ratio r (any term divided by the one before), and the number of terms n (count the exponents: from 0 to $n-1$ is n terms).
2. If $r = 1$, return na.
3. Otherwise return $a\dfrac{1-r^n}{1-r}$.
4. Sanity-check with $n = 1$ (should give a) or by adding the first few terms.

Output: the sum.

```python
def geometric_sum(a, r, n):
    """Sum of the first n terms a + ar + ... + ar^(n-1)."""
    if r == 1:
        return a * n
    return a * (1 - r**n) / (1 - r)

def brute(a, r, n):
    return sum(a * r**k for k in range(n))

for a, r, n in [(3, 2, 10), (1, 0.5, 20), (5, -0.5, 7), (2, 1, 9)]:
    print(a, r, n, geometric_sum(a, r, n), brute(a, r, n))
```

```
3 2 10 3069.0 3069
1 0.5 20 1.9999980926513672 1.9999980926513672
5 -0.5 7 3.359375 3.359375
2 1 9 18 18
```

14.5.4. Worked examples

Savings (an annuity). $a = 100$, $r = 1.05$, $n = 10$:

$$\begin{aligned} S &= 100 \cdot \frac{1.05^{10} - 1}{1.05 - 1} \\ &= 100 \cdot \frac{1.628895 - 1}{0.05} \qquad (1.05^{10} \approx 1.628895) \\ &= 100 \cdot 12.57789 \\ &\approx \$1{,}257.79. \end{aligned}$$

You deposited \$1,000; the other \$257.79 is interest on interest.

Powers of two. $1 + 2 + 4 + \cdots + 2^{n-1}$ has $a = 1, r = 2$:

$$\sum_{k=0}^{n-1} 2^k = \frac{2^n - 1}{2 - 1} = 2^n - 1.$$

So $1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 = 255 = 2^8 - 1$. Every programmer knows this one in disguise: the largest unsigned 8-bit number, binary `11111111`, is $2^8 - 1$. Notice also that the sum of all the earlier powers of two is *one less than the next power*. That fact is the engine behind the dynamic-array analysis later in the chapter.

Alternating ratio. $5 - 2.5 + 1.25 - 0.625 + \cdots$ to 7 terms: $a = 5$, $r = -0.5$, $n = 7$. $(-0.5)^7 = -0.0078125$, so

$$S_7 = 5 \cdot \frac{1 - (-0.0078125)}{1 - (-0.5)} = 5 \cdot \frac{1.0078125}{1.5} = 3.359375,$$

matching the Python output above.

Choosing a method: arithmetic or geometric?

- Terms change by a **fixed amount** → arithmetic → "number of terms × average of ends".
- Terms change by a **fixed factor** → geometric → shift and subtract.
- Neither, but the summand is a polynomial in k → split with the sigma rules and use the power-sum formulas.
- Neither and nothing familiar → compute numerically, look for telescoping (see the telescoping exercise at the end of the chapter), or bound the sum (see Section 14.7).

14.5.5. Where you'd use this

- **Mortgages and loans:** a fixed monthly payment P on a loan of L at monthly rate i over n months must satisfy $L = P\sum_{k=1}^{n}(1+i)^{-k}$: the loan equals the present value of all the payments. The geometric-series formula turns this into *Try it:* Mortgage payment and schedule .

  $$P = \frac{Li}{1-(1+i)^{-n}}.$$

  For a $200,000 mortgage at 4.8% a year ($i = 0.004$ per month) over 25 years ($n = 300$), $P \approx \$1{,}145.99$ a month. Over the full term you pay about $343,800, so about $143,800 is interest. Every bank's mortgage calculator is this one line.

- **Pensions:** the savings example above is the same formula run forwards. $200 a month for 40 years at 0.5% a month grows to $200 \cdot \frac{1.005^{480}-1}{0.005} \approx \$398{,}000$, of which only $96,000 was paid in.
- **Computing:** a complete binary tree of height h has $1 + 2 + \cdots + 2^h = 2^{h+1} - 1$ nodes, so a balanced tree holding a million keys needs only about 20 levels. And the exponential-backoff client above, after 8 failed attempts, has waited $0.1(1 + 2 + \cdots + 2^7) = 0.1(2^8 - 1) = 25.5$ s in total. That's the number you put in the timeout configuration.

14.6. Infinite geometric series

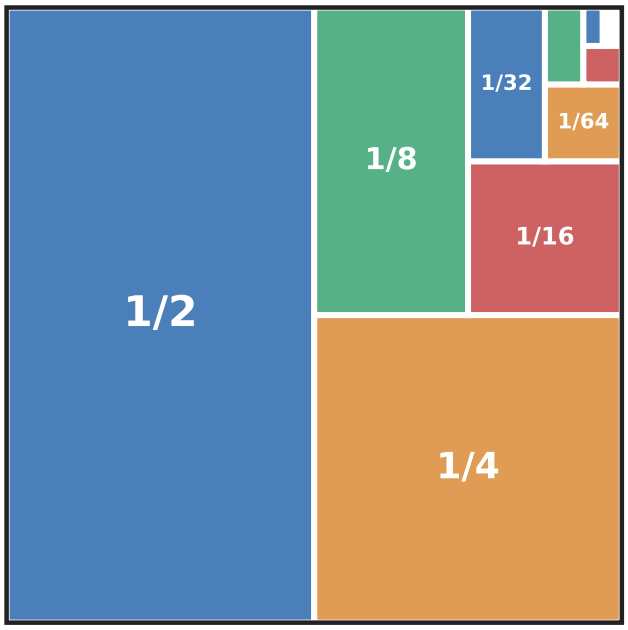
14.6.1. The problem: Zeno's paradox

Around 450 BCE Zeno of Elea argued that motion is impossible. To walk to a wall you must first cover half the distance, then half of what remains, then half of *that*, and so on — infinitely many tasks. How can infinitely many tasks ever be completed?

In numbers: the distances are $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$ of the total, and the question becomes whether

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots$$

can sensibly have a value. The picture strongly suggests the answer is 1: the pieces tile a unit square ever more completely.



$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots = 1$$

Figure 117. Halves of halves of halves fill the unit square: the infinite sum is 1.

14.6.2. Intuition: watch the partial sums

We cannot literally perform infinitely many additions. What we *can* do is look at the partial sums and see whether they settle down:

n	partial sum S_n	gap $1 - S_n$
1	0.5	0.5
2	0.75	0.25
3	0.875	0.125
5	0.96875	0.03125
10	0.9990234375	0.0009765625
20	0.99999904632…	0.00000095367…

The gap halves each time. It never becomes exactly zero, but it becomes smaller than any positive tolerance you care to name. *That* is what we mean when we say the infinite sum equals 1: not that we finish adding, but that the partial sums **converge** to 1.

14.6.3. Why it works: let n go to infinity in the finite formula

The finite formula already contains the answer. For n terms,

$$S_n = a\,\frac{1 - r^n}{1 - r}.$$

If $|r| < 1$, then r^n shrinks towards 0 as n grows (each multiplication by r makes it smaller in size). So S_n approaches $a \cdot \frac{1-0}{1-r}$:

$$\boxed{\sum_{k=0}^{\infty} ar^k = \frac{a}{1 - r}, \qquad |r| < 1.}$$

If $|r| \geq 1$, r^n does not shrink, and the partial sums either grow without bound ($r \geq 1$), alternate with growing size ($r < -1$), or bounce between two values forever ($r = -1$). In all these cases the infinite series has no value; we say it **diverges**.

The phrase "approaches as n grows" is the idea of a **limit**, made precise in Volume 2, Chapter 16 . For now the intuitive meaning — "gets and stays as close as you like" — is enough.



Convergence of a series

An infinite series $\sum_{k=1}^{\infty} a_k$ **converges** to S if its partial sums $S_n = \sum_{k=1}^{n} a_k$ get arbitrarily close to S and stay there as $n \to \infty$ (read "as n tends to infinity"). We then write $\sum_{k=1}^{\infty} a_k = S$ and call S the **sum** of the series. If the partial sums do not approach any finite number, the series **diverges**. The symbol ∞ (**infinity**) here is not a number; $\sum^{\infty}$ is shorthand for "the limit of the partial sums".

14.6.4. Worked examples

Zeno. $a = \frac{1}{2}, r = \frac{1}{2}$: $\frac{1/2}{1-1/2} = 1$. ✓

0.999… = 1. The repeating decimal $0.\overline{9}$ *means* the series

$$\frac{9}{10} + \frac{9}{100} + \frac{9}{1000} + \cdots = \sum_{k=1}^{\infty} \frac{9}{10^k}, \qquad a = \tfrac{9}{10},\ r = \tfrac{1}{10}.$$

$$0.\overline{9} = \frac{9/10}{1 - 1/10} = \frac{9/10}{9/10} = 1.$$

Not "infinitely close to 1", not "1 minus a tiny bit": exactly 1. The partial sums $0.9, 0.99, 0.999, \ldots$ fall short by $\frac{1}{10}, \frac{1}{100}, \frac{1}{1000}, \ldots$, and the only number they approach is 1.

```python
from fractions import Fraction

# Partial sums of 9/10 + 9/100 + 9/1000 + ...
s = Fraction(0)
for k in range(1, 8):
    s += Fraction(9, 10**k)
    print(k, s, 1 - s)
```

```
1 9/10 1/10
2 99/100 1/100
3 999/1000 1/1000
4 9999/10000 1/10000
5 99999/100000 1/100000
6 999999/1000000 1/1000000
7 9999999/10000000 1/10000000
```

Repeating decimals to fractions. $0.\overline{27} = 0.272727\ldots = \frac{27}{100} + \frac{27}{100^2} + \cdots$:

$$0.\overline{27} = \frac{27/100}{1 - 1/100} = \frac{27/100}{99/100} = \frac{27}{99} = \frac{3}{11}.$$

This is why every repeating decimal is a rational number (Chapter 1).

The bouncing ball, all the way. Dropped from 10 m, rebounding to 60% each time. The ball falls 10 m, then rises and falls 6 m, rises and falls 3.6 m, and so on:

$$\begin{aligned} \text{total distance} &= 10 + 2(6 + 3.6 + 2.16 + \cdots) \\ &= 10 + 2 \cdot \frac{6}{1 - 0.6} \qquad\qquad (a = 6,\ r = 0.6) \\ &= 10 + 2 \cdot 15 = 40 \text{ m}. \end{aligned}$$

Achilles and the tortoise. Achilles runs 10 times as fast as a tortoise that starts 100 m ahead. By the time Achilles reaches the tortoise's start, it has moved 10 m; when he covers that, it has moved 1 m; and so on. Achilles runs $100 + 10 + 1 + 0.1 + \cdots = \frac{100}{1 - 0.1} = 111.\overline{1}$ m before drawing level — a finite distance, covered in a finite time. Zeno's infinitely many tasks fit inside a finite total.

14.6.5. Where you'd use this

- **Medicine:** you take a 100 mg tablet every morning, and by the next morning your body has cleared half of what was there (a 24-hour half-life). Just after the nth dose the amount is $100(1 + 0.5 + 0.5^2 + \cdots + 0.5^{n-1})$: 100, 150, 175, 187.5 mg, and so on. The infinite series gives the long-run peak, $\frac{100}{1 - 0.5} = 200$ mg, and the trough just before each dose, half that, 100 mg. Pharmacists use exactly this to choose a dose and interval that keep the drug between "effective" and "toxic", and it is why some prescriptions start with a double **loading dose**: it reaches the steady level on day one.

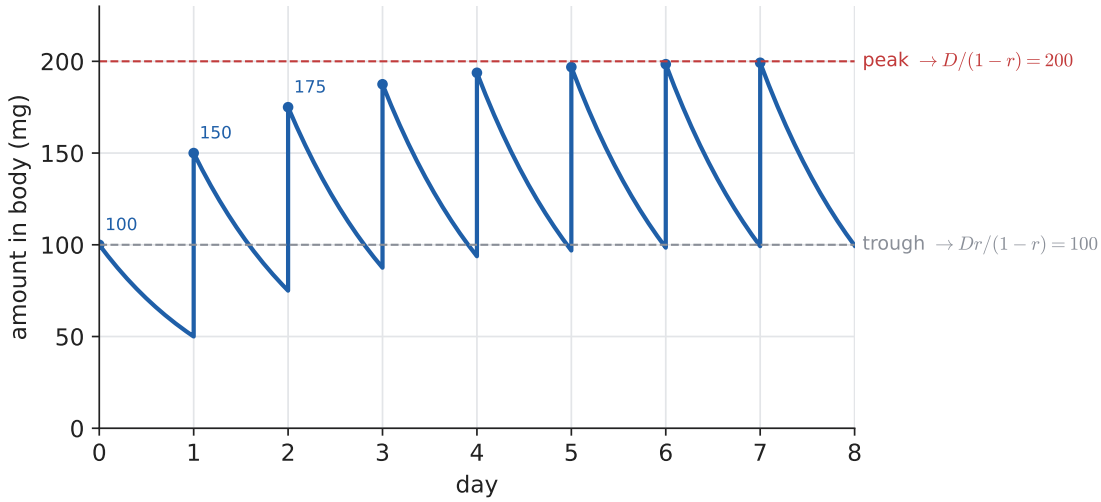


Figure 118. Daily 100 mg doses with half cleared every 24 hours. Each peak is a partial sum of a geometric series, and the peaks approach $100/(1-0.5) = 200$ *mg.*

- **Economics:** if every household spends 80% of any extra income it receives, then \$1 of new government spending becomes \$0.80 of someone's income, of which \$0.64 is spent again, and so on. The total extra spending is $1 + 0.8 + 0.64 + \cdots = \frac{1}{1-0.8} = \$5$. That's the **Keynesian multiplier**.
- **Computing:** in reinforcement learning, an agent values a reward that arrives k steps in the future at γ^k times its face value, with a **discount factor** γ such as 0.99. A reward of 1 on every step forever is worth $\sum_{k=0}^{\infty} 0.99^k = \frac{1}{1-0.99} = 100$. So $\frac{1}{1-\gamma}$ acts as the agent's **effective horizon**: with $\gamma = 0.99$ it plans roughly 100 steps ahead.



Infinite sums do not behave like finite ones

Applying $\frac{a}{1-r}$ blindly when $|r| \geq 1$ gives nonsense. With $r = 2$: "$1 + 2 + 4 + 8 + \cdots = \frac{1}{1-2} = -1$." The partial sums are 1, 3, 7, 15, … and plainly head off to infinity; the formula's derivation needed $r^n \to 0$, which is false here. Always check $|r| < 1$ *before* using the infinite formula.



Grandi's series

What is $1 - 1 + 1 - 1 + 1 - \cdots$? Grouping as $(1-1) + (1-1) + \cdots$ suggests 0; grouping as $1 + (-1+1) + (-1+1) + \cdots$ suggests 1; the formula with $r = -1$ (illegitimately) gives $\frac{1}{2}$. In 1703 Guido Grandi argued this showed God could create the world from nothing. The modern resolution: the partial sums $1, 0, 1, 0, \ldots$ never settle, so the series diverges, and regrouping an infinite sum is not a safe operation unless the series converges. (There are more exotic summation methods, used in physics, that do assign it $\frac{1}{2}$ — but that is a different definition of "sum".)



Geometric series slider

Choose a and drag a slider for r from -1.5 to 1.5. Plot the partial sums $S_1, \ldots, S_{40}$ as dots, with a horizontal line at $\frac{a}{1-r}$ when $|r| < 1$. Watch the dots zig-zag in when $r < 0$, creep in slowly when r is close to 1, and break away to infinity the moment $|r|$ reaches 1.

Try it: Partial sums, slide the ratio

14.7. Convergence and divergence: a first look

14.7.1. The problem

The geometric series is unusually friendly: we have an exact formula for its partial sums. Most series are not like that. Given $\sum a_k$, how can we tell whether it converges without a closed form? A full answer belongs to a calculus course, but two facts take you a long way, and one famous example is essential culture.

14.7.2. The divergence test: terms must shrink to zero

If a series converges, its partial sums settle down, so consecutive partial sums get very close together. But $S_n - S_{n-1} = a_n$. So **if $\sum a_k$ converges, then $a_k \to 0$.** Turned around:

$$\textbf{if the terms } a_k \textbf{ do not approach } 0, \textbf{ the series } \sum a_k \textbf{ diverges.}$$

For example, $\sum_{k=1}^{\infty} \frac{k}{k+1} = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \cdots$ diverges, since the terms approach 1: you keep adding almost 1 forever.

The test is one-directional. Terms going to zero is **necessary** but **not sufficient**, and the harmonic series is the example everyone should know.

14.7.3. The harmonic series diverges

$$\sum_{k=1}^{\infty} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots$$

The terms shrink to zero. Surely the sum converges? Compute: after 10 terms it is about 2.93; after 100, about 5.19; after a million, about 14.39. It is growing, but painfully slowly. Is there a ceiling?

There is not. The proof, due to Nicole Oresme around 1350, is a model of how to bound something you cannot compute: **group the terms into blocks that each add up to at least one half.**

$$\begin{aligned}
\sum_{k=1}^{\infty} \frac{1}{k} &= 1 + \frac{1}{2} + \left(\frac{1}{3} + \frac{1}{4}\right) + \left(\frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8}\right) + \left(\frac{1}{9} + \cdots + \frac{1}{16}\right) + \cdots \\
&\geq 1 + \frac{1}{2} + \left(\frac{1}{4} + \frac{1}{4}\right) + \left(\frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8}\right) + \left(\frac{1}{16} + \cdots + \frac{1}{16}\right) + \cdots && \text{(shrink each term to the block's last)} \\
&= 1 + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \cdots && \text{(each block now sums to exactly } \tfrac{1}{2}\text{).}
\end{aligned}$$

Block m contains the 2^{m-1} terms from $\frac{1}{2^{m-1}+1}$ to $\frac{1}{2^m}$; each is at least $\frac{1}{2^m}$, so the block is at least $2^{m-1}\cdot\frac{1}{2^m}=\frac{1}{2}$. Adding infinitely many halves exceeds every bound, so the harmonic series diverges. More precisely, the first 2^m terms sum to at least $1+\frac{m}{2}$.

$$1+\tfrac{1}{2}+\left(\tfrac{1}{3}+\tfrac{1}{4}\right)+\left(\tfrac{1}{5}+\cdots+\tfrac{1}{8}\right)+\left(\tfrac{1}{9}+\cdots+\tfrac{1}{16}\right)+\cdots$$

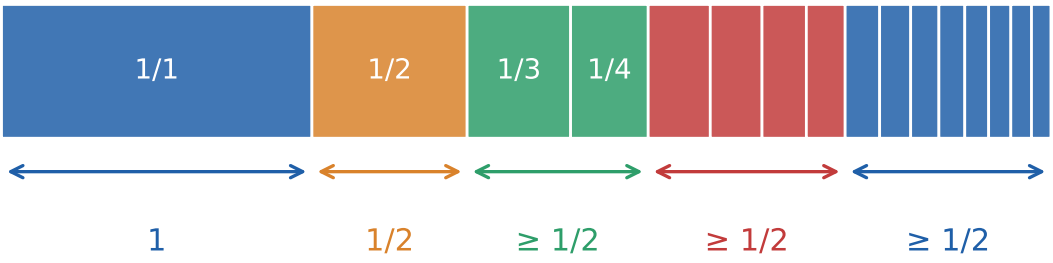


Figure 119. Grouping the harmonic series into blocks, each worth at least one half.



Grouping consecutive terms of a series of *positive* terms is safe: it just means looking at a subsequence of the partial sums. The danger shown by Grandi's series arises with terms of mixed sign.

14.7.4. How slowly?

The partial sums $H_n=\sum_{k=1}^{n}\frac{1}{k}$ (the **harmonic numbers**) grow like the natural logarithm: $H_n\approx\ln n+0.5772$, where 0.5772… is the **Euler–Mascheroni constant** γ. The reason will be clear once you know that $\int_1^n\frac{dx}{x}=\ln n$ (Volume 2, Chapter 19). To push the sum past 10 you need more than twelve thousand terms:

```python
import math

H, n, target = 0.0, 0, 10
while H < target:
    n += 1
    H += 1 / n
print(n, H)
print(math.log(n) + 0.5772156649)
```

```
12367 10.000043008275778
10.000002578642057
```

To pass 100 you would need about $e^{99.42}\approx 1.5\times10^{43}$ terms. Divergent, but only just.

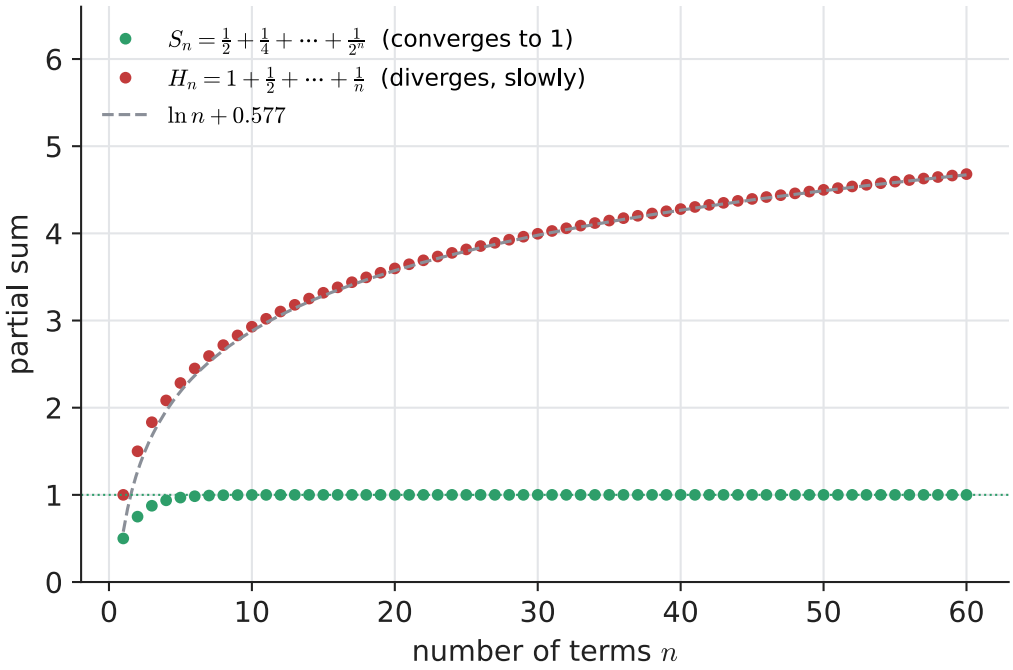


Figure 120. Partial sums of a convergent geometric series level off; harmonic partial sums keep climbing, tracking ln n.

14.7.5. Comparison: a first convergence tool

The harmonic proof used an idea worth naming. For series with **positive** terms, partial sums only go up, so there are just two possibilities: they grow without bound, or they are capped and settle to a limit. Hence the **comparison test**: if $0 \le a_k \le b_k$ for every k and $\sum b_k$ converges, then $\sum a_k$ converges too (it is capped by the same ceiling). If $a_k \ge b_k \ge 0$ and $\sum b_k$ diverges, so does $\sum a_k$.

Example. Does $\sum_{k=1}^{\infty} \frac{1}{2^k + k}$ converge? Each term is smaller than $\frac{1}{2^k}$, and $\sum \frac{1}{2^k} = 1$. So yes, and its sum is less than 1.

14.7.6. Where you'd use this

The harmonic numbers H_n turn up wherever you wait for "the last one of n".

- **Collecting stickers (the coupon-collector problem):** a sticker album has 50 different stickers, and each pack contains one at random. When you already have k of them, a new pack gives you a new one with probability $(50 - k)/50$, so on average you wait $50/(50 - k)$ packs for it. The total expected number of packs is

$$\frac{50}{50} + \frac{50}{49} + \cdots + \frac{50}{1} = 50\, H_{50} \approx 50 \times 4.499 \approx 225.$$

  Nearly a quarter of that is spent on the last four stickers ($50(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}) \approx 104$ packs). That is why the last few are so annoying, and why sticker swaps exist.

- **Physics and engineering:** stack identical books at the edge of a table, each sticking out as far as it can without toppling. With n books the top one can overhang by $\frac{1}{2} H_n$ book lengths. Four books give $\frac{1}{2}(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}) \approx 1.04$, so the top book sits entirely beyond the table edge. Because H_n

diverges, a tall enough stack could reach any distance at all, but slowly: an overhang of 3 lengths needs 227 books.

- **Computing:** the coupon-collector bound tells you how many random requests it takes before a load balancer has hit all n servers, or how many random tests are needed before every one of n code paths has been exercised: about $n \ln n$. The same $\ln n$ from harmonic sums gives the average cost of quicksort, about $2n \ln n$ comparisons.



Basel problem

Squaring the denominators changes everything: $\sum_{k=1}^{\infty} \frac{1}{k^2}$ converges. Finding its exact value defeated the best mathematicians for decades until Leonhard Euler announced in 1734 that

$$1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \cdots = \frac{\pi^2}{6}.$$

Why should π, a circle number, appear in a sum of reciprocal squares? Euler's argument treated $\frac{\sin x}{x}$ as an "infinite polynomial" with roots at $\pm\pi, \pm 2\pi, \ldots$. More generally $\sum \frac{1}{k^p}$ converges exactly when $p > 1$; the harmonic series sits right on the knife-edge $p = 1$.



When to trust a computed sum

You cannot decide convergence by computing. Summing the harmonic series in double-precision floating point eventually stalls: once H_n reaches about 34 (after roughly 2^{48} terms), adding $1/n$ changes nothing in the last bit, and a naive program would report a "limit". Conversely, a convergent series like $\sum (-1)^{k+1}/k$ (which equals $\ln 2$) converges so slowly that a million terms gives only about six correct digits. Use mathematics to decide *whether* a sum converges, and computation to estimate *what* it converges to (Volume 2, Chapter 32).

US curriculum

Finite and infinite geometric series, including the condition $|r| < 1$, are standard in **Algebra II** and **Precalculus** (Common Core HSA-SSE.B.4 covers the finite formula). Convergence tests (divergence test, comparison, integral test, ratio test) and the harmonic series belong to **AP Calculus BC**, or to Calculus II in college.

14.8. Recurrence relations

14.8.1. The problem

Many sequences arise naturally as recurrences, and for many of them no one hands you an explicit formula. The Tower of Hanoi puzzle moves n disks between three pegs, never placing a larger disk on a smaller. To move n disks you must move the top $n - 1$ out of the way, move the largest disk, and

then move the $n-1$ back on top. So the minimum number of moves h_n satisfies

$$h_1 = 1, \qquad h_n = 2h_{n-1} + 1.$$

How many moves for 64 disks? You do not want to iterate 64 times by hand, and you certainly want to know the *growth rate*.



Recurrence relation and order

A **recurrence relation** defines each term of a sequence from earlier terms. Its **order** is how far back it reaches: $a_n = 3a_{n-1}$ is first order, $F_n = F_{n-1} + F_{n-2}$ is second order. An order-k recurrence needs k **initial conditions**. **Solving** a recurrence means finding an explicit formula for a_n.

14.8.2. Method 1: unroll and spot the pattern

Substitute the recurrence into itself repeatedly:

$$\begin{aligned}
h_n &= 2h_{n-1} + 1 \\
&= 2(2h_{n-2} + 1) + 1 = 4h_{n-2} + 2 + 1 \\
&= 4(2h_{n-3} + 1) + 2 + 1 = 8h_{n-3} + 4 + 2 + 1 \\
&\;\;\vdots \\
&= 2^{n-1}h_1 + (2^{n-2} + \cdots + 2 + 1) && \text{(after } n-1 \text{ substitutions)} \\
&= 2^{n-1} + (2^{n-1} - 1) && (h_1 = 1\text{; geometric series)} \\
&= 2^n - 1.
\end{aligned}$$

Check: $h_1 = 1$, $h_2 = 3$, $h_3 = 7$. ✓ For 64 disks, $2^{64} - 1 \approx 1.8 \times 10^{19}$ moves; at one per second that is about 585 billion years. Notice how the geometric series appeared in the middle of the unrolling — this happens constantly.

14.8.3. Method 2: guess and verify

If you suspect a form, check it satisfies both the recurrence and the initial condition. Guess $h_n = 2^n - 1$: then $2h_{n-1} + 1 = 2(2^{n-1} - 1) + 1 = 2^n - 2 + 1 = 2^n - 1 = h_n$ ✓, and $h_1 = 2 - 1 = 1$ ✓. This "verify" step is essentially a proof by induction (Volume 3, Chapter 22).

Example 78. Procedure — solve a first-order recurrence by unrolling

Input: a recurrence $a_n = c\,a_{n-1} + g(n)$ and an initial value.

Steps:

1. Substitute the recurrence for a_{n-1}, then for a_{n-2}, and so on, writing out three or four levels in full.
2. Spot the pattern after j levels: typically $a_n = c^j a_{n-j} + (\text{a sum of } j \text{ terms})$.
3. Choose j so that a_{n-j} is the initial value.
4. Evaluate the remaining sum in closed form (it is often arithmetic or geometric).
5. Verify the formula on the first two or three terms.

Output: an explicit formula for a_n.

14.8.4. Fibonacci numbers

The most famous recurrence is second order:

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2} \quad (n \geq 2),$$

giving $0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots$. Leonardo of Pisa (Fibonacci) introduced it in 1202 as an idealised rabbit-breeding puzzle, but it turns up in plant spirals, in the analysis of Euclid's algorithm (Volume 3, Chapter 23), and in counting problems: F_{n+1} is the number of ways to tile a $1 \times n$ strip with squares and dominoes.

Ratios of consecutive terms. Compute F_{n+1}/F_n:
$1, 2, 1.5, 1.667, 1.6, 1.625, 1.615, 1.619, 1.6176, 1.6182, \ldots$ The ratios settle to a limit. If the ratio approaches some φ, then for large n, $F_n \approx \varphi F_{n-1}$ and $F_{n-1} \approx \varphi F_{n-2}$; substituting into the recurrence and dividing by F_{n-2}:

$$\begin{aligned} \varphi^2 F_{n-2} &\approx \varphi F_{n-2} + F_{n-2} \\ \varphi^2 &= \varphi + 1 && \text{(divide by } F_{n-2}\text{)} \\ \varphi^2 - \varphi - 1 &= 0 \\ \varphi &= \frac{1+\sqrt{5}}{2} \approx 1.618034 && \text{(quadratic formula, positive root).} \end{aligned}$$

(The quadratic formula is derived in Chapter 8.) This is the **golden ratio**. The same quadratic (called the **characteristic equation**) leads to the exact closed form, **Binet's formula**:

$$F_n = \frac{\varphi^n - \psi^n}{\sqrt{5}}, \qquad \varphi = \frac{1+\sqrt{5}}{2}, \quad \psi = \frac{1-\sqrt{5}}{2} \approx -0.618.$$

Since $|\psi| < 1$, the ψ^n term shrinks to nothing, and F_n is simply $\varphi^n/\sqrt{5}$ rounded to the nearest integer. For instance, $\varphi^{20}/\sqrt{5} \approx 6765.00003$, and $F_{20} = 6765$. The upshot: **Fibonacci numbers grow exponentially, by a factor of about 1.618 per step.** (The general technique for linear recurrences, using the characteristic equation, is best seen through eigenvalues: Volume 2, Chapter 28 .)



The golden ratio is overrated (a bit)

The golden ratio really does appear in sunflower seed heads and pine cones, where successive seeds are placed about $360°/\varphi^2 \approx 137.5°$ apart — the angle that packs them most evenly. But many popular claims (the Parthenon, the Mona Lisa, "the most pleasing rectangle") do not survive measurement. In mathematics its fame is deserved for a different reason: φ is, in a precise sense, the irrational number hardest to approximate by fractions, because its continued fraction is $1 + \cfrac{1}{1 + \cfrac{1}{1 + \cdots}}$.

14.8.5. Worked example: a recurrence from a story

A population of fish is 2,000. Each year it grows by 10% and then 150 fish are harvested. Find a formula for the population p_n after n years.

Set up. $p_0 = 2000$, $p_n = 1.1\,p_{n-1} - 150$.

Unroll.

$$\begin{aligned} p_n &= 1.1\,p_{n-1} - 150 \\ &= 1.1^2 p_{n-2} - 150(1.1) - 150 \\ &= 1.1^3 p_{n-3} - 150(1.1^2 + 1.1 + 1) \\ &\;\;\vdots \\ &= 1.1^n p_0 - 150 \sum_{k=0}^{n-1} 1.1^k \\ &= 2000\,(1.1)^n - 150 \cdot \frac{1.1^n - 1}{0.1} \qquad \text{(geometric series)} \\ &= 2000\,(1.1)^n - 1500\,(1.1)^n + 1500 \\ &= 500\,(1.1)^n + 1500. \end{aligned}$$

Check. $p_0 = 500 + 1500 = 2000$ ✓. $p_1 = 1.1(2000) - 150 = 2050$; formula: $550 + 1500 = 2050$ ✓.

Interpret. 1500 is the **equilibrium** (if $p = 1500$, then $1.1p - 150 = 1500$ again). Starting above it, the population grows exponentially away from it; starting below it, the harvest would drive the population to zero. A single closed form reveals behaviour the recurrence hides.

14.8.6. Where you'd use this

- **Credit cards and loans:** you owe \$1,000 at 2% interest a month and pay \$50 a month. The balance obeys $b_n = 1.02\,b_{n-1} - 50$ with $b_0 = 1000$. Unrolling exactly as for the fish gives $b_n = 2500 - 1500(1.02)^n$, which reaches zero when $1.02^n = \frac{5}{3}$, at $n = \ln(5/3)/\ln 1.02 \approx 25.8$, so you're clear in 26 months. The equilibrium is the warning sign: at \$20 a month the equilibrium is $20/0.02 = \$1{,}000$, your current balance, so you would pay forever and never reduce the debt.
- **Biology and ecology:** the fish model is a real tool for setting fishing quotas, and the same first-order recurrences describe drug clearance, cell cultures and invasive species.
- **Computing:** the next section is entirely about recurrences in software: loops, growable arrays, divide-and-conquer and memoisation.

14.9. Sums and recurrences in algorithm analysis

This section gathers the computing payoffs. Each one is a direct application of something above.

14.9.1. Loops are sums

The running time of a loop is the sum, over iterations, of the work done in each iteration. Consider checking every unordered pair of n items (for duplicate detection, collision tests, or a naive closest-pair search):

```python
def count_pairs(n):
    ops = 0
    for i in range(n):
        for j in range(i + 1, n):   # every unordered pair (i, j) with i < j
            ops += 1
    return ops

for n in [10, 100, 1000]:
    print(n, count_pairs(n), n * (n - 1) // 2)
```

```
10 45 45
100 4950 4950
1000 499500 499500
```

Translate the loop into a sum. For a fixed i, the inner loop runs for $j = i+1, \dots, n-1$, which is $n - 1 - i$ iterations. So

$$\begin{aligned}
\text{total} &= \sum_{i=0}^{n-1} (n - 1 - i) \\
&= (n-1) + (n-2) + \cdots + 1 + 0 && \text{(write out the terms)} \\
&= \sum_{m=0}^{n-1} m && \text{(same terms in reverse: } m = n-1-i\text{)} \\
&= \frac{(n-1)n}{2}.
\end{aligned}$$

That is $\frac{1}{2}n^2 - \frac{1}{2}n$. For large n the n^2 term dominates, so we say the loop is $O(n^2)$ ("order n squared"): doubling n roughly quadruples the work. The triangle-shaped loop does half the work of a full $n \times n$ double loop, but halving does not change the growth rate. The precise meaning of $O(\cdot)$ as a statement about limits is in Volume 2, Chapter 16.

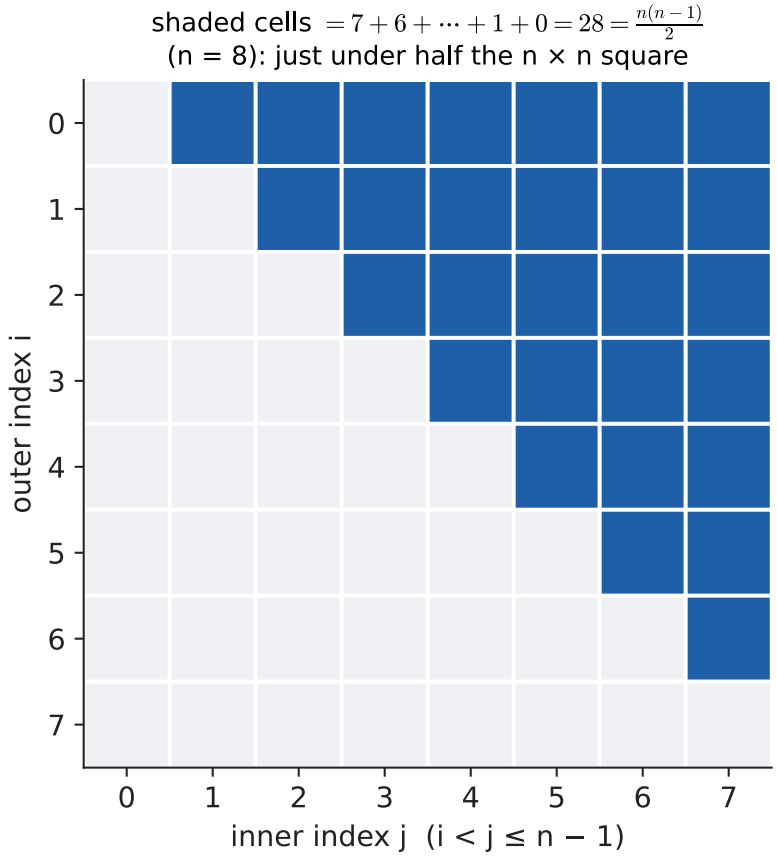


Figure 121. The loop visits the cells above the diagonal, the pairs $i < j$. For $n = 8$ that is $7 + 6 + \cdots + 1 + 0 = 28$ of the 64 cells, just under half: a triangular number.

Computer-science connection: insertion sort and the arithmetic series

Insertion sort on a reverse-sorted array of length n shifts 1 element, then 2, then 3, ..., then $n - 1$: a total of $\frac{n(n-1)}{2}$ shifts. Selection sort, bubble sort, and the naive "compare every pair" algorithms all hide the same arithmetic series. Whenever the inner loop's length grows linearly with the outer index, expect $\Theta(n^2)$. A triple loop with bounds $i < j < k$ gives $\binom{n}{3} \approx n^3/6$ — the next triangular-like number up (Volume 3, Chapter 24).

14.9.2. Dynamic arrays: why append is O(1) on average

Python lists, Java `ArrayList`, C++ `std::vector`, and Go slices all grow by **doubling**: when the array is full, allocate one twice as big and copy everything across. Copying is expensive, so how can appending be "constant time"? *Try it:* Dynamic array amortised cost.

Count element writes for n appends, starting from capacity 1. Every append writes one new element: n writes. Resizes happen when the size reaches 1, 2, 4, 8, …, copying that many elements. If the largest resize copies 2^m elements, where $2^m < n$, the total copying is a geometric series:

$$\begin{aligned} \text{copies} &= 1 + 2 + 4 + \cdots + 2^m = 2^{m+1} - 1 && \text{(geometric series)} \\ &< 2 \cdot 2^m < 2n && \text{(since } 2^m < n\text{).} \end{aligned}$$

So the total work is less than $n + 2n = 3n$: fewer than 3 writes per append **on average**, however large n grows. This "average over a sequence of operations" is called **amortised cost**. The key is the powers-of-two fact: the sum of all previous resizes is smaller than the latest one.

```python
class DynamicArray:
    def __init__(self):
        self.capacity, self.size = 1, 0
        self.data = [None]
        self.writes = 0              # count every element written

    def append(self, x):
        if self.size == self.capacity:
            self.capacity *= 2
            new = [None] * self.capacity
            for i in range(self.size):   # copy the old contents
                new[i] = self.data[i]
                self.writes += 1
            self.data = new
        self.data[self.size] = x
        self.size += 1
        self.writes += 1

for n in [10, 1000, 1_000_000]:
    arr = DynamicArray()
    for k in range(n):
        arr.append(k)
    print(n, arr.writes, round(arr.writes / n, 3))
```

```
10 25 2.5
1000 2023 2.023
1000000 2048575 2.049
```

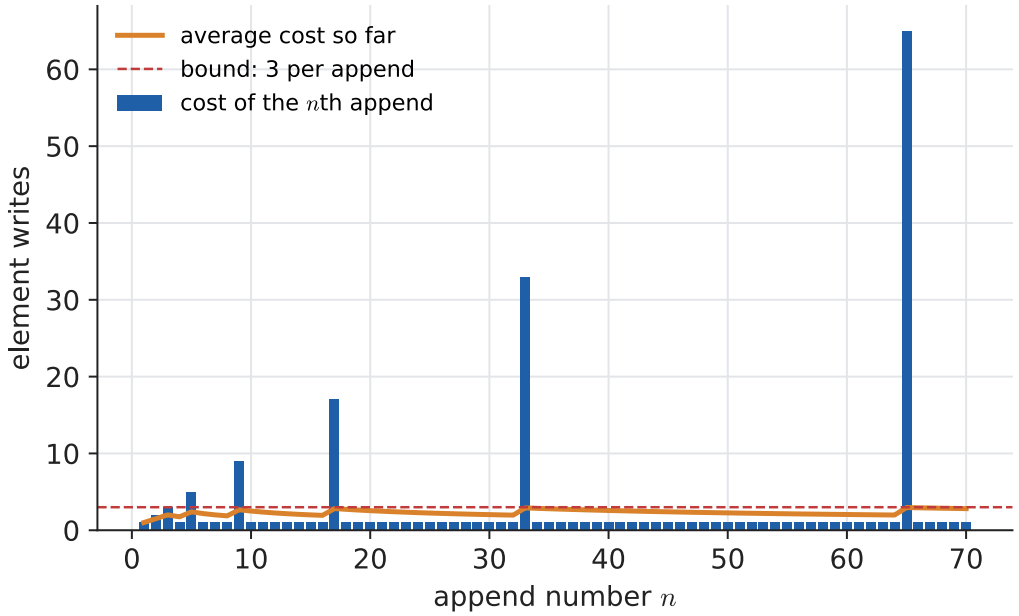


Figure 122. Individual appends occasionally cost a lot (the spikes are resizes), but the running average stays below 3.



Grow by a constant amount and you lose

Suppose instead the array grows by a fixed 100 slots each time. Resizes copy 100, 200, 300, … elements — an *arithmetic* series. After n appends there have been about $n/100$ resizes, costing about $100 \cdot \frac{(n/100)^2}{2} = \frac{n^2}{200}$ copies: $O(n^2)$ total, $O(n)$ per append on average. Arithmetic growth leads to quadratic cost; geometric growth leads to linear cost. Any factor greater than 1 works (CPython uses about 1.125, many C++ libraries 1.5 or 2), trading wasted memory for fewer copies.

Try it: Per-append cost, step by step ; Grow by 100: quadratic cost .

14.9.3. Divide-and-conquer recurrences

Merge sort splits an array in half, sorts each half recursively, and merges the halves in linear time. Its cost satisfies, informally (for n a power of 2, measuring cost in units where merging n items costs n), *Try it:* Merge-sort recursion tree .

$$T(1) = 0, \qquad T(n) = 2\,T(n/2) + n.$$

Unroll it exactly as we did for Hanoi:

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &= 2\big(2T(n/4) + n/2\big) + n = 4T(n/4) + 2n \\ &= 4\big(2T(n/8) + n/4\big) + 2n = 8T(n/8) + 3n \\ &\;\;\vdots \\ &= 2^k\,T(n/2^k) + kn \qquad\qquad\qquad\qquad\qquad\qquad \text{(after } k \text{ levels).} \end{aligned}$$

The recursion bottoms out when $n/2^k = 1$, i.e. $k = \log_2 n$ (Chapter 12). Then

$$T(n) = nT(1) + n\log_2 n = n\log_2 n.$$

Picture a recursion tree: $\log_2 n$ levels, each doing n total work. That is where the ubiquitous $O(n \log n)$ comes from. Compare the recurrence $T(n) = T(n/2) + 1$ (binary search: one half discarded, constant work), which unrolls to $\log_2 n$, and $T(n) = 2T(n/2) + 1$ (visiting every node of a balanced tree), which unrolls to a geometric series $1 + 2 + 4 + \cdots + n/2 \approx n$. The general pattern is captured by the "master theorem" in algorithms textbooks; behind it are nothing but geometric series.

PYTHON

```python
import math

def T(n):
    """T(n) = 2 T(n/2) + n, T(1) = 0, for n a power of 2."""
    if n == 1:
        return 0
    return 2 * T(n // 2) + n

for k in [1, 4, 10, 20]:
    n = 2**k
    print(n, T(n), n * int(math.log2(n)))
```

Try it: Binary search recurrence ; Visiting every tree node .

```
2 2 2
16 64 64
1024 10240 10240
1048576 20971520 20971520
```

14.9.4. Fibonacci: recursion versus memoisation

Translate the Fibonacci recurrence literally into code and you get something correct but catastrophically slow:

```python
from functools import lru_cache

calls = 0
def fib_naive(n):
    global calls
    calls += 1
    if n < 2:
        return n
    return fib_naive(n - 1) + fib_naive(n - 2)

@lru_cache(maxsize=None)
def fib_memo(n):
    if n < 2:
        return n
    return fib_memo(n - 1) + fib_memo(n - 2)

def fib_iter(n):
    a, b = 0, 1
    for _ in range(n):
        a, b = b, a + b
    return a

print(fib_naive(25), calls)
print(fib_memo(25), fib_memo.cache_info().misses)
print(fib_iter(100))
```

```
75025 242785
75025 26
354224848179261915075
```

Why 242,785 calls? Let C_n be the number of calls made by `fib_naive(n)`. Then $C_0 = C_1 = 1$ and $C_n = 1 + C_{n-1} + C_{n-2}$ — the call count obeys a Fibonacci-like recurrence itself. Its solution is $C_n = 2F_{n+1} - 1$ (check: $2F_{26} - 1 = 2(121{,}393) - 1 = 242{,}785$). So the cost grows like $\varphi^n \approx 1.618^n$: exponential. The call tree shows why: the same subproblems are recomputed again and again.

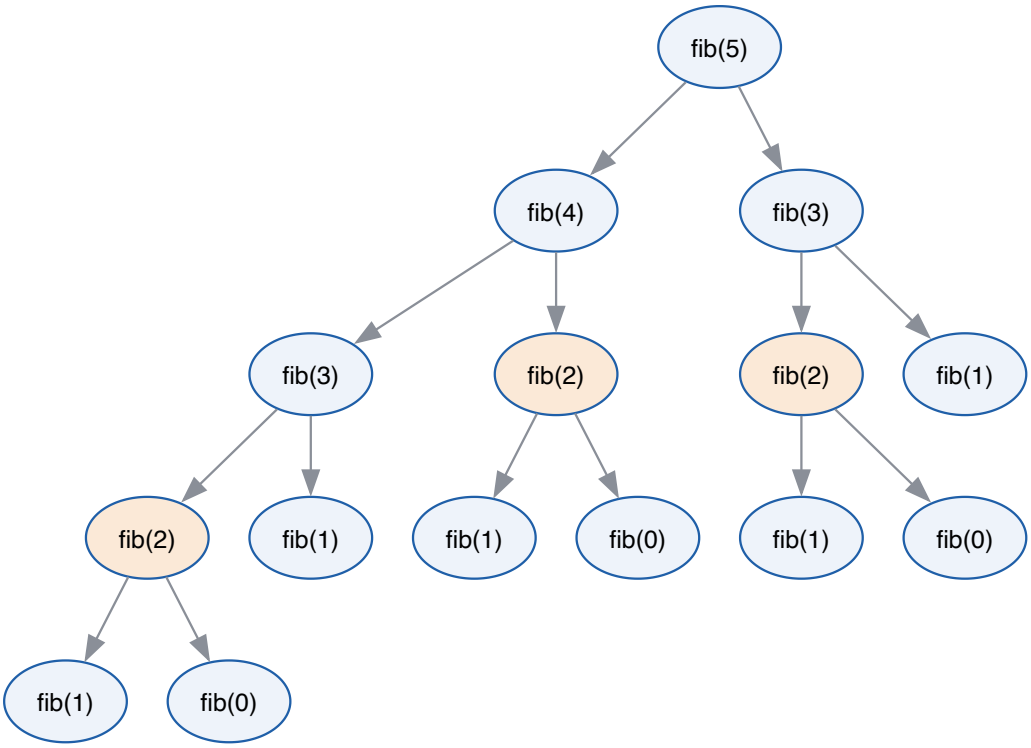


Figure 123. The call tree of naive fib(5). fib(2) (shaded) is computed three times, fib(3) twice; the waste compounds exponentially.

Memoisation caches each result the first time it is computed, so each F_k is computed once: 26 distinct computations for $n = 25$, linear in n. The iterative version goes further and keeps only the last two values — the recurrence evaluated "bottom up". That move from top-down recursion with a cache to bottom-up iteration over a table is the essence of **dynamic programming**. *Try it:* Call tree of fib(5) .

Computer-science connection: recurrences everywhere

- **Algorithm analysis:** recursive algorithms produce recurrences for their running time; solving them (by unrolling, recursion trees, or the master theorem) gives the $O(\cdot)$ bound.
- **Dynamic programming:** edit distance, knapsack, shortest paths, and sequence alignment are all recurrences evaluated with memoisation.
- **Signal processing and finance:** an exponential moving average $s_n = \alpha x_n + (1-\alpha)s_{n-1}$ is a first-order recurrence; unrolling shows it weights past inputs by a geometric sequence $\alpha(1-\alpha)^k$.
- **Pseudo-random numbers:** a linear congruential generator $x_{n+1} = (a x_n + c) \bmod m$ is a recurrence (Volume 3, Chapter 23).
- **Machine learning:** recurrent neural networks are, literally, learned recurrence relations $h_n = f(h_{n-1}, x_n)$.

Visualise the waste

Animate the call tree of `fib(n)` for n up to 10, colouring nodes by argument. Toggle "memoise" and watch the tree collapse into a single path with short side-branches. Display call counts side by side with $2F_{n+1} - 1$ and with $n + 1$.

Try it: Memoised fib collapses · Visualise the waste · Count the calls in Python

Gauss's staircase, live

Drag a slider for n and watch two staircases of 1 to n blocks slide together into an $n \times (n + 1)$ rectangle. A second mode shows the nested-loop grid filling cell by cell as a program runs, with a running counter matching $\frac{n(n-1)}{2}$.

Try it: Staircase against n(n + 1)/2 · Count the loop in Python

US curriculum

Recursive definitions appear in **Algebra I** and **Algebra II**; the Fibonacci sequence is a common enrichment topic. Solving recurrences, the analysis of algorithms, and dynamic programming are college material — typically **Discrete Mathematics** and **Data Structures and Algorithms** (or **AP Computer Science A** for the loop-counting ideas in a lighter form).

14.10. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Loan and mortgage calculator
- Graphing calculator
- Big-O explorer and benchmark
- Language sandbox

14.11. Exercises

Recognition

1. Classify each as arithmetic, geometric, or neither, and give a_n if possible: (a) $4, 1, -2, -5, \ldots$ (b) $48, 12, 3, \frac{3}{4}, \ldots$ (c) $1, 4, 9, 16, \ldots$ (d) $2, -6, 18, -54, \ldots$.
2. Write out every term and evaluate: (a) $\sum_{k=1}^{4} k^3$ (b) $\sum_{j=0}^{3} (-2)^j$ (c) $\prod_{i=1}^{4} \frac{i}{i+1}$ (d) $\sum_{i=5}^{3} i$.
3. Write in sigma notation: (a) $3 + 6 + 9 + \cdots + 300$ (b) $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots - \frac{1}{99}$.

Application

4. Find the sum of all three-digit odd numbers.
5. A job offer starts at \$60,000 with a 3% raise every year. What are total earnings over the first 20 years? (Give the closed form and a numerical value.)
6. **Telescoping.** Use $\frac{1}{k(k+1)} = \frac{1}{k} - \frac{1}{k+1}$ to evaluate $\sum_{k=1}^{n} \frac{1}{k(k+1)}$, and hence the infinite sum.
7. Convert $0.4\overline{6} = 0.4666\ldots$ and $1.\overline{142857}$ into fractions.
8. Does each converge? If so, find the sum. (a) $\sum_{k=0}^{\infty} 5\left(-\frac{2}{3}\right)^k$ (b) $\sum_{k=1}^{\infty} \left(\frac{5}{4}\right)^k$ (c) $\sum_{k=1}^{\infty} \frac{2k}{3k+1}$ (d) $\sum_{k=1}^{\infty} \frac{1}{3^k+1}$.

Problem solving

9. Solve the recurrence $a_0 = 5$, $a_n = 3a_{n-1} - 4$ by unrolling. Verify for $n = 1, 2$.
10. How many times does the innermost statement run, as an exact function of n?

```python
for i in range(n):
    for j in range(i):
        for k in range(j):
            work()
```

11. Show that $\sum_{k=1}^{\infty} \frac{1}{2k}$ diverges using the harmonic series, and explain why $\sum_{k=1}^{\infty} \frac{1}{k \cdot 2^k}$ converges.
12. A recursive algorithm satisfies $T(1) = 1, T(n) = 3T(n/3) + n$ for n a power of 3. Unroll to find $T(n)$ exactly.

▼ **Answers**

1. (a) Arithmetic, $d = -3$: $a_n = 4 - 3(n-1) = 7 - 3n$. (b) Geometric, $r = \frac{1}{4}$: $a_n = 48\left(\frac{1}{4}\right)^{n-1}$. (c) Neither (differences 3, 5, 7; ratios 4, 2.25, 1.78): $a_n = n^2$. (d) Geometric, $r = -3$: $a_n = 2(-3)^{n-1}$.
2. (a) $1 + 8 + 27 + 64 = 100$ (and $100 = (\frac{4 \cdot 5}{2})^2$, as the sum-of-cubes formula predicts). (b) $1 - 2 + 4 - 8 = -5$. (c) $\frac{1}{2} \cdot \frac{2}{3} \cdot \frac{3}{4} \cdot \frac{4}{5} = \frac{1}{5}$ (everything cancels except the first numerator and last denominator). (d) Upper bound below lower bound: empty sum, 0.
3. (a) $\sum_{k=1}^{100} 3k$. (b) The denominators are odd: $2k - 1$ for $k = 1, \ldots, 50$ (since $2 \cdot 50 - 1 = 99$); signs alternate starting positive: $\sum_{k=1}^{50} \frac{(-1)^{k+1}}{2k-1}$.
4. Odd three-digit numbers: $101, 103, \ldots, 999$. $d = 2$, $n = \frac{999-101}{2} + 1 = 450$. $S = \frac{450(101+999)}{2} = 450 \times 550 = 247{,}500$.
5. Geometric with $a = 60000$, $r = 1.03$, $n = 20$: $S = 60000 \cdot \frac{1.03^{20} - 1}{0.03}$. $1.03^{20} \approx 1.806111$, so $S \approx 60000 \times 26.8704 \approx \$1{,}612{,}222$.
6. $\sum_{k=1}^{n} \left(\frac{1}{k} - \frac{1}{k+1}\right) = \left(1 - \frac{1}{2}\right) + \left(\frac{1}{2} - \frac{1}{3}\right) + \cdots + \left(\frac{1}{n} - \frac{1}{n+1}\right)$. Each negative term cancels the next positive term ("telescopes"), leaving $1 - \frac{1}{n+1} = \frac{n}{n+1}$. As $n \to \infty$ this approaches 1, so the infinite sum is 1.

7. $0.4\overline{6} = \frac{4}{10} + \left(\frac{6}{100} + \frac{6}{1000} + \cdots\right) = \frac{4}{10} + \frac{6/100}{1-1/10} = \frac{4}{10} + \frac{6}{90} = \frac{36}{90} + \frac{6}{90} = \frac{42}{90} = \frac{7}{15}$. $1.\overline{142857} = 1 + \frac{142857/10^6}{1-10^{-6}} = 1 + \frac{142857}{999999} = 1 + \frac{1}{7} = \frac{8}{7}$.

8. (a) $|r| = \frac{2}{3} < 1$: converges to $\frac{5}{1+2/3} = \frac{5}{5/3} = 3$. (b) $r = \frac{5}{4} > 1$: diverges. (c) Terms approach $\frac{2}{3} \neq 0$: diverges by the divergence test. (d) $0 < \frac{1}{3^k+1} < \frac{1}{3^k}$ and $\sum_{k\geq 1} 3^{-k} = \frac{1/3}{1-1/3} = \frac{1}{2}$: converges by comparison, to something less than $\frac{1}{2}$.

9. $a_n = 3a_{n-1} - 4 = 9a_{n-2} - 4(3+1) = \cdots = 3^n a_0 - 4(3^{n-1} + \cdots + 3 + 1) = 5 \cdot 3^n - 4 \cdot \frac{3^n-1}{2} = 5 \cdot 3^n - 2 \cdot 3^n + 2 = 3^{n+1} + 2$. Check: $a_1 = 3 \cdot 5 - 4 = 11 = 9 + 2$ ✓; $a_2 = 3 \cdot 11 - 4 = 29 = 27 + 2$ ✓. (Equilibrium is $a = 3a - 4 \Rightarrow a = 2$, visible in the formula.)

10. For fixed i and j, the inner loop runs j times. Summing over $j < i$: $\sum_{j=0}^{i-1} j = \frac{i(i-1)}{2}$. Then summing over i: $\sum_{i=0}^{n-1} \frac{i(i-1)}{2}$. This counts triples $k < j < i$ chosen from $\{0, \ldots, n-1\}$, which is $\binom{n}{3} = \frac{n(n-1)(n-2)}{6}$. (Check $n = 4$: $i = 2$ contributes 1, $i = 3$ contributes 3, total 4 = $\binom{4}{3}$ ✓.) It is $O(n^3)$.

11. $\sum \frac{1}{2k} = \frac{1}{2}\sum \frac{1}{k}$; half of something that grows without bound also grows without bound, so it diverges. For the second, $0 < \frac{1}{k2^k} \leq \frac{1}{2^k}$ and $\sum \frac{1}{2^k}$ converges (to 1), so it converges by comparison. (Its sum is in fact $\ln 2$.)

12. $T(n) = 3T(n/3) + n = 9T(n/9) + 2n = \cdots = 3^k T(n/3^k) + kn$. With $k = \log_3 n$: $T(n) = n \cdot T(1) + n\log_3 n = n + n\log_3 n$. Check $n = 3$: $3T(1) + 3 = 6 = 3 + 3 \cdot 1$ ✓.

14.12. Mastery check

1. Find the 40th term and the sum of the first 40 terms of $-3, 2, 7, 12, \ldots$.
2. Evaluate $\sum_{k=1}^{20}(4k-1)$ using the sigma rules (show the split).
3. Derive the formula for $1 + r + r^2 + \cdots + r^{n-1}$ by shift-and-subtract, then use it to compute $\sum_{k=0}^{9} 3^k$.
4. Express $0.\overline{81}$ as a fraction in lowest terms.
5. Explain in two or three sentences why $\sum \frac{1}{k}$ diverges even though $\frac{1}{k} \to 0$.
6. A loop `for i in range(1, n+1): for j in range(i, n+1): work()` — exactly how many calls to `work()`?
7. Solve $a_1 = 2$, $a_n = a_{n-1} + 3n$. (Hint: unroll to a sum and use the sigma rules.)

Decision rule: All correct without hesitation → move on to Chapter 15 (or Volume 2, Chapter 16 if complex numbers are already solid). Stuck on 1–2 → reread Section 14.4 and Section 14.5. Stuck on 5 → reread Section 14.7. Stuck on 6–7 → work through Section 14.9 and Section 14.8, then the section exercises.

▼ **Mastery check answers**

1. $a_1 = -3$, $d = 5$: $a_{40} = -3 + 39 \cdot 5 = 192$. $S_{40} = \frac{40(-3+192)}{2} = 20 \times 189 = 3780$.

2. $\sum_{k=1}^{20}(4k-1) = 4\sum_{k=1}^{20} k - \sum_{k=1}^{20} 1 = 4\cdot\frac{20\cdot 21}{2} - 20 = 840 - 20 = 820$.

3. $S = 1 + r + \cdots + r^{n-1}$, $rS = r + \cdots + r^n$; subtract: $S - rS = 1 - r^n$, so $S = \frac{1-r^n}{1-r}$ for $r \neq 1$. With $r = 3$, $n = 10$: $\frac{3^{10}-1}{3-1} = \frac{59049-1}{2} = 29{,}524$.

4. $\frac{81/100}{1-1/100} = \frac{81}{99} = \frac{9}{11}$.

5. Group the terms into blocks $\left(\frac{1}{3}+\frac{1}{4}\right), \left(\frac{1}{5}+\cdots+\frac{1}{8}\right), \ldots$; block m has 2^{m-1} terms each at least $1/2^m$, so each block is at least $\frac{1}{2}$. Infinitely many blocks each worth at least $\frac{1}{2}$ add up to more than any bound. The terms shrink, but not fast enough: you keep finding another half.

6. For each i, the inner loop runs $n - i + 1$ times.
   $\sum_{i=1}^{n}(n-i+1) = n + (n-1) + \cdots + 1 = \frac{n(n+1)}{2}$.

7. $a_n = a_1 + \sum_{k=2}^{n} 3k = 2 + 3\left(\frac{n(n+1)}{2} - 1\right) = \frac{3n(n+1)}{2} - 1$. Check: $a_1 = 3 - 1 = 2$ ✓, $a_2 = 2 + 6 = 8 = 9 - 1$ ✓, $a_3 = 8 + 9 = 17 = 18 - 1$ ✓.

Chapter 15. Complex Numbers

Chapter map: the equation $x^2 + 1 = 0$ has no real solution. Inventing a number i with $i^2 = -1$ looks like a cheat, yet it produces a number system in which every polynomial equation can be solved, multiplication becomes rotation, and trigonometry, exponentials, and geometry turn out to be one subject. This chapter builds complex numbers from the arithmetic up to Euler's formula, De Moivre's theorem, and roots of unity, then shows them at work in circuits, signals, graphics, the FFT, and the Mandelbrot set.

Why this chapter matters: you will never pay for groceries with $3 + 2i$, but you use complex numbers every day without seeing them. Your phone's 4G and Wi-Fi signals encode data as points in the complex plane, the mains electricity in your house is designed with phasors, every MP3 and JPEG went through a Fourier transform built from roots of unity, and every game engine that rotates a sprite is multiplying by $e^{i\theta}$ under another name. Complex numbers are the natural language for anything that rotates or oscillates, and they turn pages of trigonometry into one line of algebra.

Where this chapter sits

You will need: the number system up to the reals (Chapter 1), algebraic manipulation and expanding brackets (Chapter 4), polynomials and the quadratic formula (Chapter 7, Chapter 8), coordinates and distance (Chapter 10), the exponential function (Chapter 12), and sine, cosine, radians, and the angle-addition formulas (Chapter 13). Series (Chapter 14) help with one optional argument.

This chapter unlocks: solving any polynomial equation in principle; the eigenvalues of rotation matrices (Volume 2, Chapter 28); oscillations and differential equations; Fourier analysis and the FFT; and a much deeper understanding of trigonometry.

15.1. The problem: equations with no solutions

15.1.1. Two stories

The modern version. Solve $x^2 + 1 = 0$. Rearranging gives $x^2 = -1$. But the square of any real number is non-negative: $3^2 = 9$, $(-3)^2 = 9$, $0^2 = 0$. No real number squares to -1. The parabola $y = x^2 + 1$ floats entirely above the x-axis, never touching it. In Chapter 8 this is the case of a **negative discriminant**, and we said "no real roots". The word *real* was doing a lot of work there.

The historical version. Complex numbers were *not* invented to solve $x^2 + 1 = 0$. Mathematicians were perfectly happy to say that equation had no solution, just as $x + 5 = 2$ once "had no solution" before negative numbers were accepted. What forced the issue was the **cubic**. In 1545 Gerolamo Cardano published a formula for solving cubics. Applied to

$$x^3 = 15x + 4,$$

the formula produces

$$x = \sqrt[3]{2 + \sqrt{-121}} + \sqrt[3]{2 - \sqrt{-121}}.$$

Square roots of negative numbers! Yet this equation obviously has a real solution: $x = 4$ works, because $4^3 = 64 = 15 \cdot 4 + 4$. Around 1572 Rafael Bombelli had what he called "a wild thought". He treated $\sqrt{-1}$ as a symbol obeying ordinary algebra plus the rule $(\sqrt{-1})^2 = -1$, and noticed that

$$\begin{aligned}
(2 + \sqrt{-1})^3 &= (2 + \sqrt{-1})^2 \, (2 + \sqrt{-1}) \\
&= (4 + 4\sqrt{-1} - 1)(2 + \sqrt{-1}) \qquad \text{(using } (\sqrt{-1})^2 = -1\text{)} \\
&= (3 + 4\sqrt{-1})(2 + \sqrt{-1}) \\
&= 6 + 3\sqrt{-1} + 8\sqrt{-1} - 4 \\
&= 2 + 11\sqrt{-1} = 2 + \sqrt{-121}.
\end{aligned}$$

So $\sqrt[3]{2 + \sqrt{-121}} = 2 + \sqrt{-1}$, and similarly $\sqrt[3]{2 - \sqrt{-121}} = 2 - \sqrt{-1}$. Adding, the imaginary parts cancel and $x = 4$. The "impossible" numbers were a necessary bridge between a real question and a real answer. That is the lesson of this chapter in miniature: complex numbers are a tool for getting real answers.

15.1.2. Intuition: extending the number line again

You have done this before. Natural numbers cannot solve $x + 5 = 2$, so we invented negatives. Integers cannot solve $3x = 2$, so we invented fractions. Rationals cannot solve $x^2 = 2$, so we filled in the irrationals (Chapter 1). Each time, the new numbers were initially distrusted ("negative", "irrational", and now "imaginary" are all slightly insulting names) and each time the extension kept all the old rules working.

The difference this time is geometric. The real numbers fill a line. There is no room left on the line for a number whose square is negative. So the new numbers must live *off* the line — in a plane.

15.2. The imaginary unit and complex arithmetic

15.2.1. Definitions



Imaginary unit

The **imaginary unit** i is a number with the defining property $i^2 = -1$. (Electrical engineers write j, since i means current; so does Python.) The square roots of a negative number $-a$ (with $a > 0$) are $\pm i\sqrt{a}$; for example $\sqrt{-9} = 3i$.



Complex number, real part, imaginary part

A **complex number** is an expression $z = a + bi$ where a and b are real numbers. a is the **real part**, written $\mathrm{Re}(z)$; b is the **imaginary part**, written $\mathrm{Im}(z)$. Note that the imaginary part is the real number b, not bi. The set of all complex numbers is written $\mathbb{C}$. Two complex numbers are equal exactly when their real parts are equal *and* their imaginary parts are equal. A number with $b = 0$ is real; a number with $a = 0, b \neq 0$ is called **purely imaginary**.

The form $a + bi$ is called **rectangular** (or **Cartesian**, or **algebraic**) form. "Complex" here means "composite" — made of two parts — not "complicated".

15.2.2. Arithmetic: ordinary algebra plus one rule

The whole of complex arithmetic is: **treat i as an unknown, do ordinary algebra, and replace i^2 by -1 whenever it appears.**

Addition and subtraction combine real parts and imaginary parts separately, like collecting like terms:

$$(a + bi) + (c + di) = (a + c) + (b + d)i, \qquad (a + bi) - (c + di) = (a - c) + (b - d)i.$$

Multiplication expands the brackets (the distributive law, Chapter 4):

$$\begin{aligned} (a + bi)(c + di) &= ac + adi + bci + bd\,i^2 && \text{(expand)} \\ &= ac + adi + bci - bd && (i^2 = -1) \\ &= (ac - bd) + (ad + bc)i && \text{(collect real and imaginary parts).} \end{aligned}$$

Do not memorise that last line; just expand and simplify each time.

Worked example. Compute $(2 + 3i)(4 - i)$. *Try it:* Multiply (2 + 3i)(4 − i) visually .

$$\begin{aligned} (2 + 3i)(4 - i) &= 8 - 2i + 12i - 3i^2 \\ &= 8 + 10i - 3(-1) \\ &= 11 + 10i. \end{aligned}$$

15.2.3. Powers of i

Since $i^2 = -1$, the powers of i cycle with period 4:

$$i^0 = 1, \quad i^1 = i, \quad i^2 = -1, \quad i^3 = i^2 \cdot i = -i, \quad i^4 = (i^2)^2 = 1, \quad i^5 = i, \ldots$$

To compute i^n, divide n by 4 and keep the remainder (Volume 3, Chapter 23): $i^{2026} = i^{4 \cdot 506 + 2} = i^2 = -1$. Negative powers too: $i^{-1} = \frac{1}{i} = \frac{i}{i^2} = -i$ (multiply top and bottom by i).

The cycle of four is a strong hint about geometry: something that returns to where it started after four steps sounds like a quarter-turn. We will confirm that shortly.



Square roots of negatives do not obey the usual rule

For positive reals $\sqrt{a}\sqrt{b} = \sqrt{ab}$. Apply it to negatives and you get nonsense:

$$-1 = i \cdot i = \sqrt{-1} \cdot \sqrt{-1} \overset{?}{=} \sqrt{(-1)(-1)} = \sqrt{1} = 1.$$

The faulty step is the "?". The rule $\sqrt{a}\sqrt{b} = \sqrt{ab}$ is only guaranteed for non-negative a, b. The safe habit: convert $\sqrt{-a}$ to $i\sqrt{a}$ *first*, then compute. So $\sqrt{-4} \cdot \sqrt{-9} = 2i \cdot 3i = 6i^2 = -6$.

Computer-science connection: Python's complex type

Python has complex numbers built in, written with the engineers' `j`. The `cmath` module provides complex versions of the `math` functions.

PYTHON

```
import cmath

z = 3 + 4j          # Python uses j, the engineers' letter
w = 1 - 2j
print(z + w, z - w, z * w, z / w)
print(z.real, z.imag, z.conjugate(), abs(z))
print(cmath.phase(z))            # argument, in radians
print(cmath.polar(z))            # (modulus, argument)
print(cmath.rect(2, cmath.pi / 3))   # back from polar form
print(1j ** 2)
```

```
(4+2j) (2+6j) (11-2j) (-1+2j)
3.0 4.0 (3-4j) 5.0
0.9272952180016122
(5.0, 0.9272952180016122)
(1.0000000000000002+1.7320508075688772j)
(-1+0j)
```

Internally a complex number is just a pair of floats; NumPy arrays of dtype `complex128` store two `float64`s per element. C has `double complex`, C++ `std::complex`, and Fortran has had a `COMPLEX` type since the early 1960s — scientific computing has always needed them. Every function in this chapter (`abs`, `phase`, `polar`, `rect`, `exp`) is in the output above or in later examples.

15.3. The complex plane

15.3.1. Intuition: a number is a point

A complex number $a + bi$ is determined by two real numbers, so it can be pictured as the point (a, b) in a plane. The horizontal axis holds the real numbers (the **real axis**); the vertical axis holds the purely imaginary numbers (the **imaginary axis**). This picture is the **complex plane**, or **Argand diagram** (after Jean-Robert Argand, 1806, although Caspar Wessel published it first in 1799).

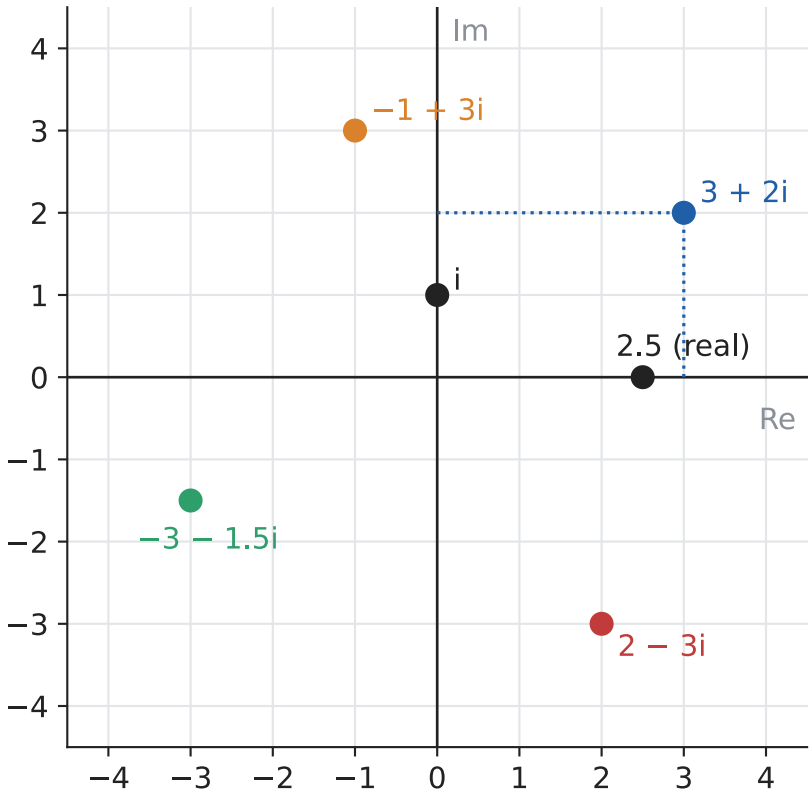


Figure 124. The complex plane. Each complex number a + bi is the point (a, b). Real numbers sit on the horizontal axis; i is one unit up.

The real number line is still there, intact, as the horizontal axis. Complex numbers did not replace the reals; they surround them.

Addition now has a familiar picture: adding complex numbers is adding vectors (Volume 2, Chapter 26), tip to tail. Subtraction $z - w$ is the arrow from w to z, so $|z - w|$ (defined next) is the **distance** between them. *Try it:* Add complex numbers as vectors .

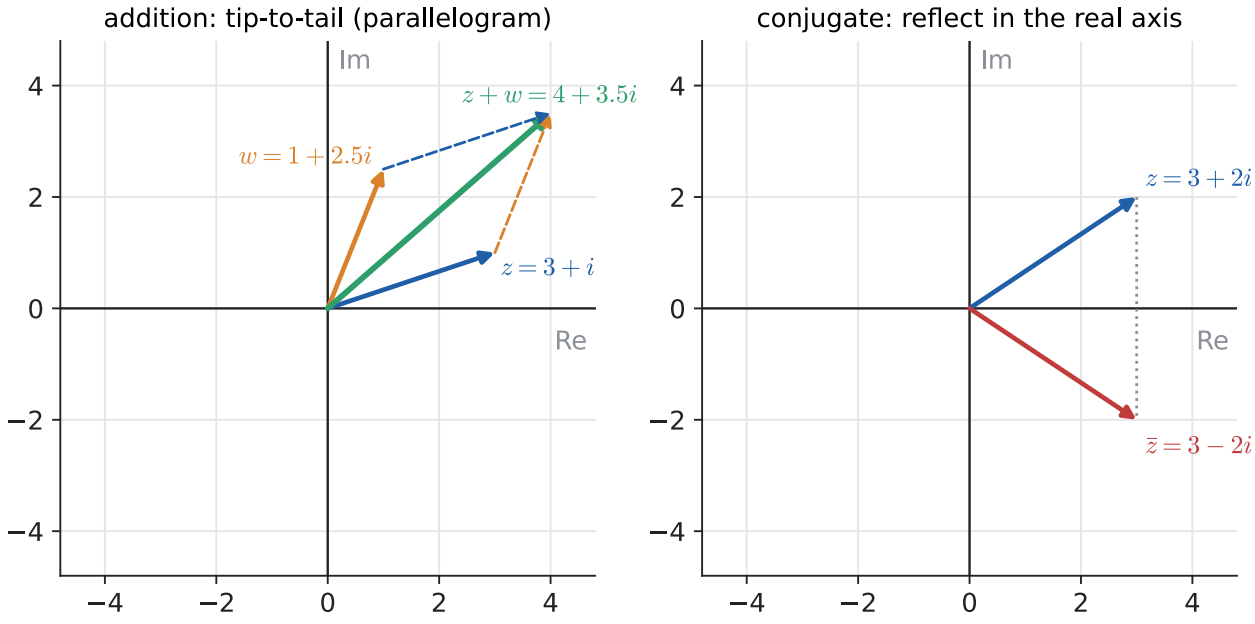


Figure 125. Left: adding complex numbers is vector addition. Right: the conjugate is the mirror image in the real axis.

15.3.2. Modulus and argument

A point in the plane can be described by its coordinates (a, b) or by its **distance** from the origin and **direction** (Chapter 13). For complex numbers these get special names.

Modulus and argument

The **modulus** (or **absolute value**, or **magnitude**) of $z = a + bi$ is its distance from 0: $|z| = \sqrt{a^2 + b^2}$ (Pythagoras). For a real number it agrees with the ordinary absolute value.

The **argument** $\arg z$ is the angle from the positive real axis to the arrow for z, measured anticlockwise, usually in radians. It is defined only for $z \neq 0$ and only up to adding multiples of 2π. The **principal argument** is the choice in $(-\pi, \pi]$.

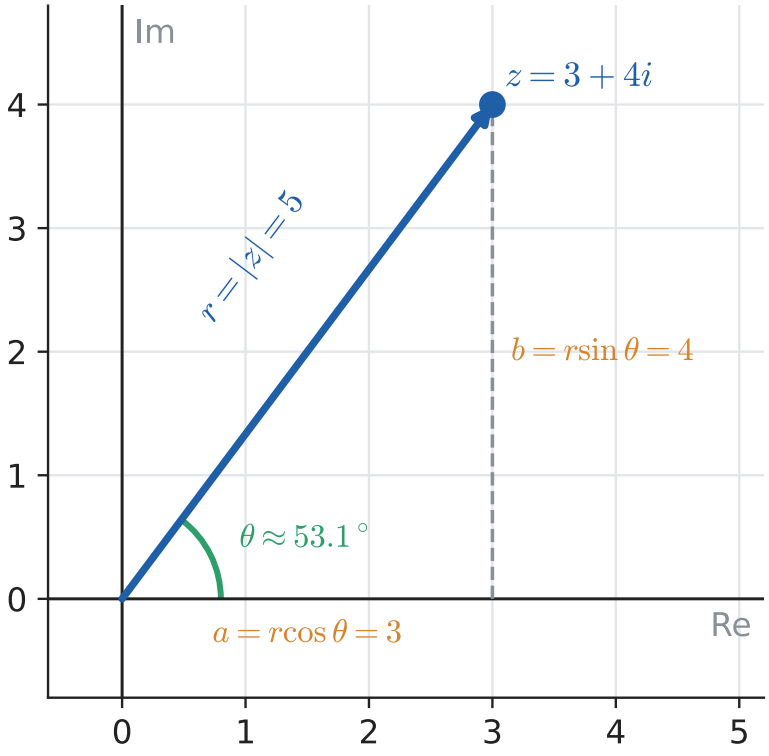


Figure 126. The modulus r and argument θ of z = 3 + 4i. The real and imaginary parts are r cos θ and r sin θ.

For $z = 3 + 4i$: $|z| = \sqrt{9 + 16} = 5$, and $\tan\theta = \frac{4}{3}$, so $\theta = \arctan\frac{4}{3} \approx 0.9273$ rad $\approx 53.13°$.

15.3.3. The method: finding the argument safely

Example 79. Procedure — modulus and argument of a + bi

Input: a nonzero complex number $z = a + bi$.

Steps:

1. Modulus: $r = \sqrt{a^2 + b^2}$.
2. Plot the point roughly, to see which quadrant it lies in.
3. Compute the reference angle $\alpha = \arctan\left|\frac{b}{a}\right|$ (if $a = 0$, the point is on the imaginary axis: $\theta = \pm\frac{\pi}{2}$).
4. Adjust for the quadrant: first quadrant $\theta = \alpha$; second $\theta = \pi - \alpha$; third $\theta = -(\pi - \alpha)$; fourth $\theta = -\alpha$.
5. Check: $r\cos\theta$ should equal a and $r\sin\theta$ should equal b.

Output: $r = |z|$ and $\theta = \arg z \in (-\pi, \pi]$.

In code, steps 3–4 are exactly `math.atan2(b, a)`, which is what `cmath.phase` uses.

Worked example. Find the modulus and argument of $z = -1 - i$. *Try it:* Find arg(−1 − i) safely.

1. $r = \sqrt{(-1)^2 + (-1)^2} = \sqrt{2}$.
2. The point $(-1, -1)$ is in the third quadrant.
3. Reference angle $\alpha = \arctan 1 = \frac{\pi}{4}$.
4. Third quadrant: $\theta = -(\pi - \frac{\pi}{4}) = -\frac{3\pi}{4}$ (that is, $-135°$).
5. Check: $\sqrt{2}\cos(-\frac{3\pi}{4}) = \sqrt{2}\cdot(-\frac{\sqrt{2}}{2}) = -1$ ✓, and the same for the sine ✓.



arctan(b/a) is in the wrong quadrant half the time

For $-1 - i$, a calculator's $\arctan\frac{b}{a} = \arctan\frac{-1}{-1} = \arctan 1 = \frac{\pi}{4}$ — pointing into the first quadrant, exactly opposite the true direction. The ratio b/a has thrown away the signs of a and b separately. Always sketch the point, or use the two-argument `atan2(b, a)`, which exists in every programming language for precisely this reason.

Try it: See arctan(b/a) point wrong.

15.3.4. Where you'd use this

The complex plane earns its keep because $|z - w|$ is the distance between two points and adding is moving.

- **Your phone's signal:** 4G, 5G and Wi-Fi transmit each group of bits as one complex number, a point in the plane called a **symbol**. In **16-QAM** the 16 allowed points are $a + bi$ with $a, b \in \{-3, -1, 1, 3\}$, so each symbol carries 4 bits. Noise shifts the received point slightly. If $2.8 - 1.2i$ arrives, the receiver decodes it as the nearest allowed point, here $3 - i$, at distance $|(2.8 - 1.2i) - (3 - i)| = |-0.2 - 0.2i| \approx 0.28$. The next-nearest points, $1 - i$ and $3 - 3i$, are about 1.8 away. Faster Wi-Fi standards pack in more points (1024-QAM carries 10 bits per symbol), which only works when there is less noise. *Try it:* Decode a 16-QAM symbol .
- **Computing:** storing 2D positions as complex numbers turns geometry into arithmetic, and Python's built-in `complex` type is a free 2D vector. Moving is adding, distance is `abs(z - w)`, and direction is `cmath.phase(z - w)`. Programmers solving grid puzzles often store a position as `z` and a heading as `d`. Then "step forward" is `z += d`, "turn left" is `d *= 1j` and "turn right" is `d *= -1j`, with no `if` statements at all.

15.4. Conjugates and division

15.4.1. The problem

Addition, subtraction, and multiplication are straightforward. What is $\frac{3+4i}{1-2i}$? We need to put it in the form $a + bi$, which means getting rid of i in the denominator. This is the same problem as rationalising a denominator like $\frac{1}{1+\sqrt{2}}$, and the same trick solves it.



Complex conjugate

The **complex conjugate** of $z = a + bi$ is $\bar{z} = a - bi$ (also written z^*, especially in physics). Geometrically it is the reflection of z in the real axis. Key facts:

- $z\bar{z} = (a + bi)(a - bi) = a^2 - abi + abi - b^2 i^2 = a^2 + b^2 = |z|^2$ — always a non-negative *real* number.
- $z + \bar{z} = 2a = 2\,\mathrm{Re}(z)$ and $z - \bar{z} = 2bi$.
- Conjugation respects arithmetic: $\overline{z + w} = \bar{z} + \bar{w}$, $\overline{zw} = \bar{z}\,\bar{w}$.

15.4.2. The method

Example 80. Procedure — divide complex numbers

Input: $\dfrac{z}{w}$ with $w = c + di \neq 0$.

Steps:

1. Multiply numerator and denominator by $\bar{w} = c - di$ (this is multiplying by 1, so the value is unchanged).
2. The denominator becomes $w\bar{w} = c^2 + d^2$, a real number.
3. Expand the numerator $z\bar{w}$.
4. Divide its real and imaginary parts by $c^2 + d^2$.

Output: $\dfrac{z}{w}$ in the form $a + bi$.

Worked example. $\dfrac{3+4i}{1-2i}$. *Try it:* Divide by the conjugate .

$$\begin{aligned}
\frac{3+4i}{1-2i} &= \frac{(3+4i)(1+2i)}{(1-2i)(1+2i)} && \text{(multiply by the conjugate of the denominator)} \\
&= \frac{3+6i+4i+8i^2}{1^2+2^2} && \text{(expand; } w\bar{w} = c^2+d^2\text{)} \\
&= \frac{3+10i-8}{5} \\
&= \frac{-5+10i}{5} = -1+2i.
\end{aligned}$$

Check by multiplying back: $(-1+2i)(1-2i) = -1+2i+2i-4i^2 = -1+4i+4 = 3+4i$ ✓.

Reciprocal. The same idea gives a formula for $1/z$:

$$\frac{1}{z} = \frac{\bar{z}}{z\bar{z}} = \frac{\bar{z}}{|z|^2}, \qquad \text{e.g.} \quad \frac{1}{3+4i} = \frac{3-4i}{25} = 0.12 - 0.16i.$$

So every nonzero complex number has a reciprocal, and $\mathbb{C}$ is a **field**: all four arithmetic operations work, with the usual laws.



No ordering

There is no sensible way to say which of two complex numbers is "bigger". Statements like $3i > 2$ or $1 + i < 2$ are meaningless; any attempt to define $<$ compatibly with arithmetic fails (if $i > 0$ then $i \cdot i > 0$, i.e. $-1 > 0$; if $i < 0$, multiplying by the negative i flips the inequality and gives the same contradiction). You *can* compare moduli: $|3i| = 3 > 2 = |2|$. When a programming language lets you sort complex numbers, it is imposing some convention, such as lexicographic order on (a, b), not a mathematical fact.

15.5. Multiplication is rotation and scaling

This is the central insight of the chapter. Once you see it, complex numbers stop being a trick and become geometry.

15.5.1. The problem

Addition has a picture (vector addition). What does *multiplication* do to points in the plane? Multiplying by a positive real like 2 stretches every arrow to twice its length. Multiplying by -1 flips every arrow to point the opposite way — a half-turn. What does multiplying by i do?

15.5.2. Intuition: multiplying by i is a quarter-turn

Take any $z = a + bi$ and multiply by i:

$$i\,(a + bi) = ai + bi^2 = -b + ai.$$

The point (a, b) has moved to $(-b, a)$. That is exactly a rotation by $90°$ anticlockwise about the origin (try $(1, 0) \to (0, 1) \to (-1, 0) \to (0, -1) \to (1, 0)$). Now the rule $i^2 = -1$ has a meaning anyone can see: **two quarter-turns make a half-turn.** And "the square root of -1" means "the operation that, done twice, reverses direction".

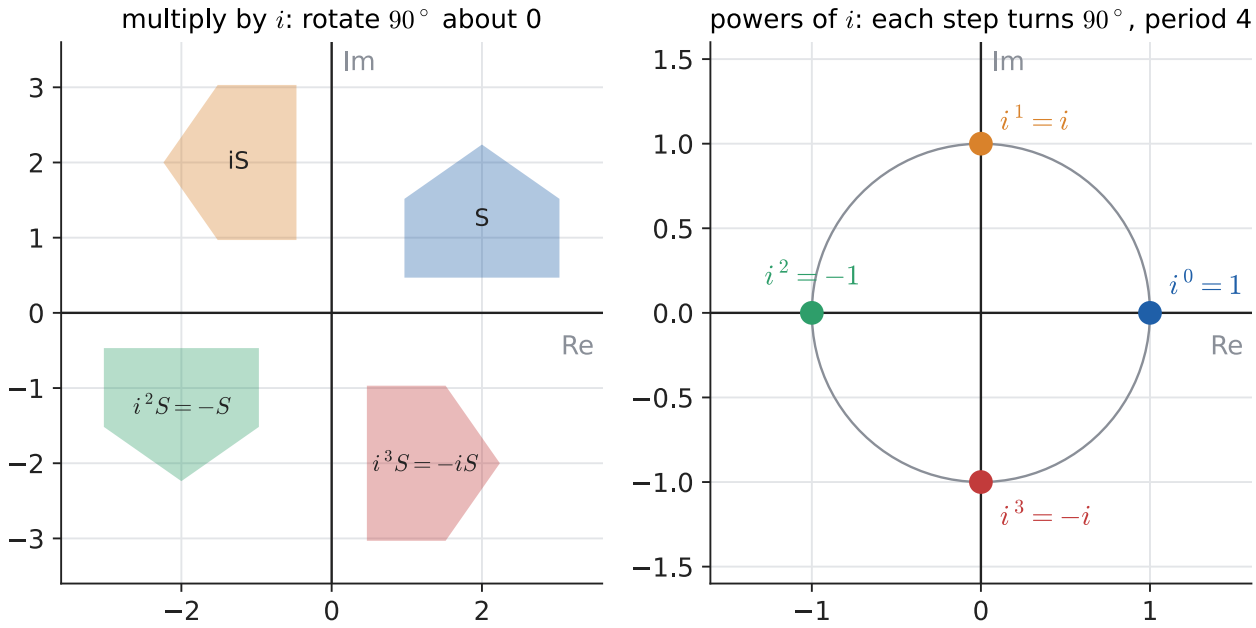


Figure 127. Left: multiplying every point of a shape by i rotates it 90° about the origin. Right: the powers of i walk around the unit circle.

15.5.3. Why it works in general

Now multiply two general complex numbers. Watch the moduli and arguments in an example: $z = 2 + i$ and $w = 1 + i$. *Try it:* Watch moduli multiply, arguments add .

$$zw = (2+i)(1+i) = 2 + 2i + i + i^2 = 1 + 3i.$$

	modulus	argument
$z = 2 + i$	$\sqrt{5} \approx 2.236$	$\approx 26.57^\circ$
$w = 1 + i$	$\sqrt{2} \approx 1.414$	45°
$zw = 1 + 3i$	$\sqrt{10} \approx 3.162$	$\approx 71.57^\circ$

The moduli *multiplied* ($\sqrt{5} \cdot \sqrt{2} = \sqrt{10}$) and the arguments *added* ($26.57^\circ + 45^\circ = 71.57^\circ$).

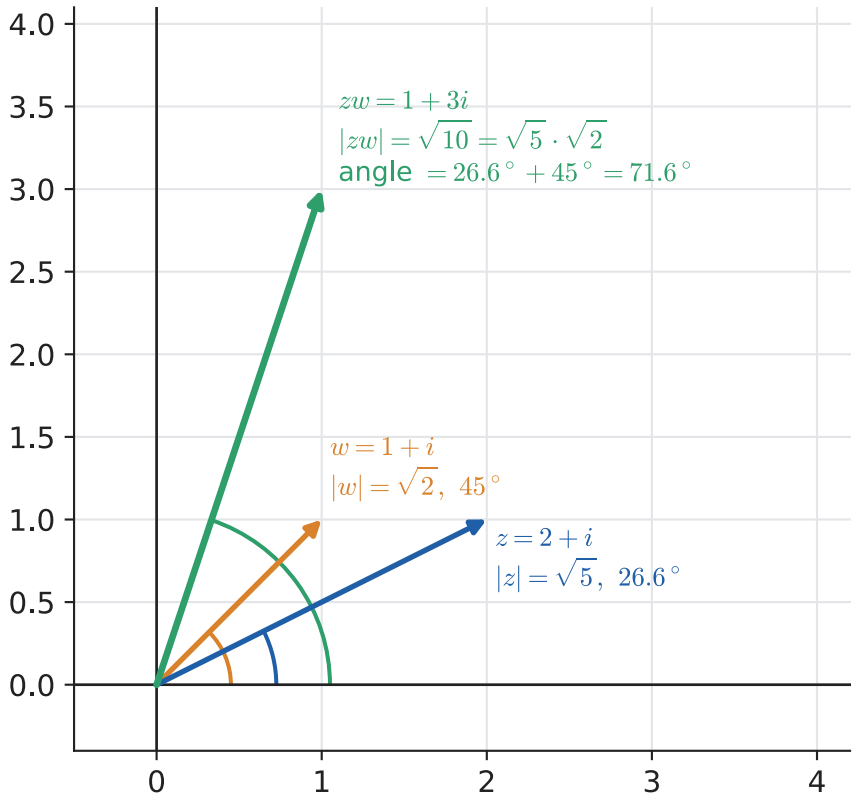


Figure 128. Multiplying z by w scales by $|w|$ and rotates by arg w: lengths multiply, angles add.

To see that this always happens, write each number using its modulus and argument (from the polar-form picture, $a = r\cos\theta$, $b = r\sin\theta$):

$$z = r(\cos\alpha + i\sin\alpha), \qquad w = s(\cos\beta + i\sin\beta).$$

Multiply:

$$\begin{aligned} zw &= rs\,(\cos\alpha + i\sin\alpha)(\cos\beta + i\sin\beta) \\ &= rs\,\big[\cos\alpha\cos\beta + i\cos\alpha\sin\beta + i\sin\alpha\cos\beta + i^2\sin\alpha\sin\beta\big] && \text{(expand)} \\ &= rs\,\big[(\cos\alpha\cos\beta - \sin\alpha\sin\beta) + i(\sin\alpha\cos\beta + \cos\alpha\sin\beta)\big] && \text{(collect; } i^2 = -1\text{)} \\ &= rs\,\big[\cos(\alpha+\beta) + i\sin(\alpha+\beta)\big] && \text{(angle-addition formulas).} \end{aligned}$$

The last step uses the angle-addition formulas from Chapter 13. So:

$$\boxed{|zw| = |z|\,|w|, \qquad \arg(zw) = \arg z + \arg w.}$$

Multiplying by w rotates by $\arg w$ and scales by $|w|$. Dividing by w does the reverse: $|z/w| = |z|/|w|$, $\arg(z/w) = \arg z - \arg w$.

Read that derivation backwards and something remarkable appears: the angle-addition formulas, which you may once have memorised as arbitrary identities, are nothing more than the real and imaginary parts of "angles add when you multiply". If you forget them, multiply out $(\cos\alpha + i\sin\alpha)(\cos\beta + i\sin\beta)$ and read them off.

Computer-science connection: 2D rotation with one multiplication

To rotate a point (x, y) by angle θ about the origin, treat it as $p = x + yi$ and multiply by the unit complex number $w = \cos\theta + i\sin\theta$:

$$(x + yi)(\cos\theta + i\sin\theta) = (x\cos\theta - y\sin\theta) + i(x\sin\theta + y\cos\theta).$$

The real and imaginary parts are exactly the rows of the 2D rotation matrix you will meet in Volume 2, Chapter 27. To rotate about a centre c instead of the origin: translate, rotate, translate back, $c + (p - c)w$.

```python
import cmath, math

def rotate(p, degrees, centre=0j):
    """Rotate point p (a complex number) about centre by the given angle."""
    w = cmath.exp(1j * math.radians(degrees))   # unit complex number
    return centre + (p - centre) * w

def tidy(z, places=4):
    return complex(round(z.real, places) + 0.0, round(z.imag, places) + 0.0)

square = [0, 1, 1 + 1j, 1j]
print([tidy(rotate(p, 90)) for p in square])
print([tidy(rotate(p, 30)) for p in square])
print(tidy(rotate(3 + 1j, 180, centre=2 + 1j)))
```

```
[0j, 1j, (-1+1j), (-1+0j)]
[0j, (0.866+0.5j), (0.366+1.366j), (-0.5+0.866j)]
(1+1j)
```

(`cmath.exp(1j * θ)` is $\cos\theta + i\sin\theta$; the next section explains why.) Composing rotations is multiplying unit complex numbers, which is cheaper and more numerically stable than multiplying matrices, and easy to renormalise if rounding errors creep in: divide by `abs(w)`. Game engines use exactly this idea one dimension up: **quaternions**, a four-dimensional extension of complex numbers invented by William Rowan Hamilton in 1843, represent 3D rotations without the "gimbal lock" of Euler angles.

15.5.4. Where you'd use this

- **Robotics and navigation:** a robot drives 3 m east, turns 60° left and drives 2 m. Each leg is a length times a unit complex number for the heading, so its position is $3 + 2e^{i\pi/3} = 3 + (1 + \sqrt{3}\,i) = 4 + 1.732i$: about 4.36 m from the start, in direction $\arctan(1.732/4) \approx 23.4°$. A turn is a multiplication and a drive is an addition, so a whole route is a short loop. Dead reckoning on ships and wheel odometry on robot vacuums work exactly like this.
- **Graphics:** to spin a sprite by 30° every frame, precompute $w = e^{i\pi/6}$ once and multiply every vertex by w: four real multiplications and two additions per vertex. After 12 frames the sprite is back where it started, because $w^{12} = e^{2\pi i} = 1$.

15.6. Polar form

The rotation picture suggests describing complex numbers by modulus and argument from the start.



Polar form

The **polar form** (or **trigonometric form**, or **modulus–argument form**) of a nonzero complex number is $z = r(\cos\theta + i\sin\theta)$, where $r = |z|$ and $\theta = \arg z$. It is often abbreviated $r\,\mathrm{cis}\,\theta$ ("r cis theta": **c**os + **i** sin), and, after the next section, written $re^{i\theta}$. Conversion: $a = r\cos\theta$, $b = r\sin\theta$; $r = \sqrt{a^2+b^2}$, $\theta = \operatorname{atan2}(b, a)$.

15.6.1. Choosing a form

- **Adding or subtracting?** Use rectangular form: add components.
- **Multiplying, dividing, raising to powers, taking roots?** Use polar form: multiply moduli, add arguments.
- **Need a final answer as $a + bi$?** Convert back at the end.

This is the same trade-off as between coordinate systems in geometry: choose the representation in which your operation is simple. Converting has a cost (a square root and an arctangent), so for a single multiplication rectangular form is usually quicker; for z^{20}, polar wins overwhelmingly.

Worked example. Write $z = 1 + \sqrt{3}\,i$ in polar form, and convert $4\,\mathrm{cis}\,\frac{2\pi}{3}$ to rectangular form. *Try it:* Convert 1 + √3 i to polar .

$$|1+\sqrt{3}i| = \sqrt{1+3} = 2, \qquad \arg = \arctan\sqrt{3} = \tfrac{\pi}{3}\ \text{(first quadrant)}, \qquad \text{so } z = 2\,\mathrm{cis}\,\tfrac{\pi}{3}.$$

$$4\,\mathrm{cis}\,\tfrac{2\pi}{3} = 4\cos\tfrac{2\pi}{3} + 4i\sin\tfrac{2\pi}{3} = 4\left(-\tfrac{1}{2}\right) + 4i\left(\tfrac{\sqrt{3}}{2}\right) = -2 + 2\sqrt{3}\,i.$$

15.7. Euler's formula

15.7.1. The problem

We found that numbers of the form $\mathrm{cis}\,\theta = \cos\theta + i\sin\theta$ obey

$$\mathrm{cis}\,\alpha \cdot \mathrm{cis}\,\beta = \mathrm{cis}(\alpha+\beta), \qquad \mathrm{cis}\,0 = 1.$$

Where have you seen a function that turns addition of inputs into multiplication of outputs? The exponential function: $e^a e^b = e^{a+b}$ and $e^0 = 1$ (Chapter 12, Chapter 3). This suggests that $\mathrm{cis}\,\theta$ is some kind of exponential of θ. Which one?

15.7.2. Intuition: three converging arguments

Argument 1 — the law of exponents. As just observed, **cis** satisfies the same law as an exponential.

Argument 2 — rates of change (a preview of Volume 2, Chapter 17). The function $e^{k\theta}$ has the property that its rate of change is k times itself. What is the rate of change of $\mathrm{cis}\,\theta$ as θ increases? The point $\mathrm{cis}\,\theta$ moves around the unit circle at unit speed, and its velocity is always perpendicular to its

position, pointing anticlockwise — that is, velocity = position rotated by $90°$ = $i\times$ position. So $\cis\theta$ changes at a rate i times itself, exactly like $e^{k\theta}$ with $k=i$. (With derivatives: $\frac{d}{d\theta}(\cos\theta+i\sin\theta)=-\sin\theta+i\cos\theta=i(\cos\theta+i\sin\theta)$.)

Argument 3 — infinite series (a preview of Taylor series, Volume 2, Chapter 18). The exponential function has a power-series expansion, an "infinite polynomial" in the sense of Chapter 14:

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

Assume this still makes sense with $x=i\theta$. Powers of $i\theta$ cycle through $1, i, -1, -i$ times powers of θ:

$$\begin{aligned} e^{i\theta} &= 1+i\theta+\frac{(i\theta)^2}{2!}+\frac{(i\theta)^3}{3!}+\frac{(i\theta)^4}{4!}+\frac{(i\theta)^5}{5!}+\cdots \\ &= 1+i\theta-\frac{\theta^2}{2!}-i\frac{\theta^3}{3!}+\frac{\theta^4}{4!}+i\frac{\theta^5}{5!}-\cdots && (i^2=-1,\ i^3=-i,\ i^4=1,\ldots) \\ &= \left(1-\frac{\theta^2}{2!}+\frac{\theta^4}{4!}-\cdots\right)+i\left(\theta-\frac{\theta^3}{3!}+\frac{\theta^5}{5!}-\cdots\right) && \textbf{(separate real and imaginary parts).} \end{aligned}$$

The two brackets are precisely the power series of $\cos\theta$ and $\sin\theta$, as you will see in Volume 2, Chapter 18 . All three arguments point to the same conclusion.

15.7.3. Euler's formula

$$\boxed{e^{i\theta}=\cos\theta+i\sin\theta}$$

Strictly speaking, this is how we *define* the exponential of an imaginary number — but it is the only definition consistent with the law of exponents, the rate-of-change property, and the power series all at once, which is why it is universally accepted. Every complex number can now be written compactly as

$$z=re^{i\theta},\qquad\text{and}\qquad re^{i\alpha}\cdot se^{i\beta}=rs\,e^{i(\alpha+\beta)}.$$

The multiplication rule is now just the law of exponents.

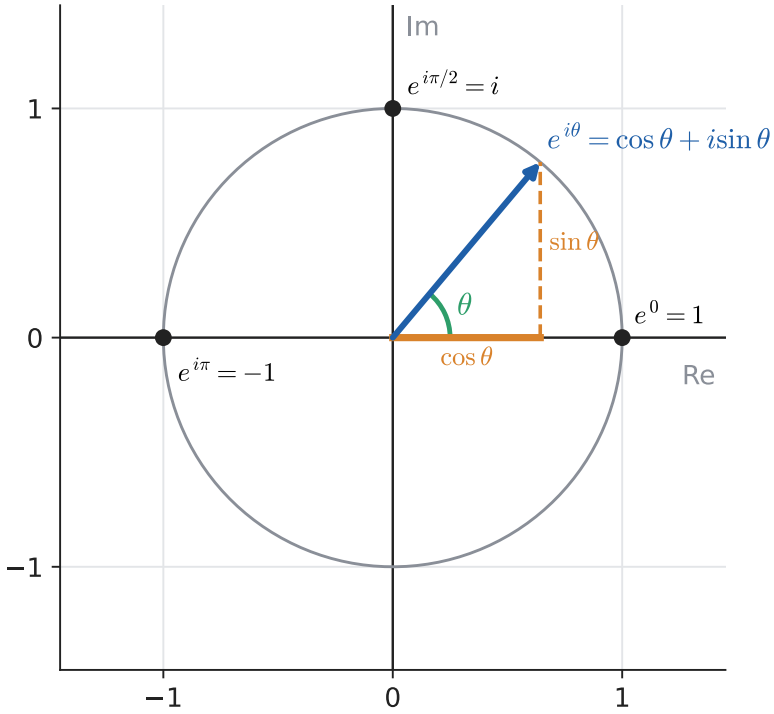


Figure 129. e^{iθ} is the point at angle θ on the unit circle; its real part is cos θ and its imaginary part is sin θ.

15.7.4. A numerical check

Trust, but verify. Compare Python's complex exponential with cosine and sine directly, and sum the power series by hand:

```python
import cmath, math

for theta in [0.0, 0.5, 1.0, math.pi / 2, math.pi]:
    lhs = cmath.exp(1j * theta)
    rhs = complex(math.cos(theta), math.sin(theta))
    print(f"{theta:8.5f}  {lhs:.12f}  {rhs:.12f}")

print(cmath.exp(1j * math.pi) + 1)
```

```
 0.00000  1.000000000000+0.000000000000j  1.000000000000+0.000000000000j
 0.50000  0.877582561890+0.479425538604j  0.877582561890+0.479425538604j
 1.00000  0.540302305868+0.841470984808j  0.540302305868+0.841470984808j
 1.57080  0.000000000000+1.000000000000j  0.000000000000+1.000000000000j
 3.14159  -1.000000000000+0.000000000000j  -1.000000000000+0.000000000000j
1.2246467991473532e-16j
```

PYTHON

```python
import math

def exp_series(z, terms=30):
    total, term = 0, 1
    for n in range(terms):
        total += term
        term = term * z / (n + 1)   # next term: z^(n+1) / (n+1)!
    return total

theta = 1.0
print(exp_series(1j * theta))
print(complex(math.cos(theta), math.sin(theta)))
```

```
(0.5403023058681397+0.8414709848078965j)
(0.5403023058681398+0.8414709848078965j)
```

The tiny $1.22 \times 10^{-16} i$ in the first output is not a flaw in Euler's formula: `math.pi` is only the closest double to π, and $\sin$ of that number is about 1.22×10^{-16} (Volume 2, Chapter 32).



Euler's identity

Setting $\theta = \pi$ gives $e^{i\pi} = -1$, usually written

$$e^{i\pi} + 1 = 0,$$

linking the five most important constants in mathematics ($0, 1, e, i, \pi$) with one each of addition, multiplication, and exponentiation. Geometrically it just says: walking π radians around the unit circle takes you halfway round, to -1. Euler's formula also yields the startling fact that i^i is a real number: $i = e^{i\pi/2}$, so $i^i = e^{i \cdot i\pi/2} = e^{-\pi/2} \approx 0.2079$. (Complex powers are multi-valued, so this is the *principal* value.)

Try it: Walk to e^(iπ) = −1 .



Euler's formula explorer

Drag θ from 0 to 2π. Show the point $e^{i\theta}$ travelling around the unit circle, with its shadow on the real axis tracing $\cos\theta$ and its shadow on the imaginary axis tracing $\sin\theta$ as two graphs unrolling beside the circle. A second panel shows the partial sums of the power series $\sum (i\theta)^n/n!$ as a chain of arrows, each turned $90°$ from the last and shrinking, spiralling into the point $e^{i\theta}$.

Try it: Open the Euler formula explorer · e^(iθ) and its shadows

15.8. De Moivre's theorem and powers

15.8.1. The idea

If multiplying by z rotates by θ and scales by r, then multiplying by z n times rotates by $n\theta$ and scales by r^n:

$$\big(r(\cos\theta + i\sin\theta)\big)^n = r^n(\cos n\theta + i\sin n\theta), \qquad \text{equivalently} \qquad (re^{i\theta})^n = r^n e^{in\theta}.$$

This is **De Moivre's theorem** (Abraham de Moivre, 1707). With Euler's formula it is just the law $(e^a)^n = e^{na}$.

Example 81. Procedure — raise a complex number to a power

Input: $z = a + bi$ and a positive integer n.

Steps:

1. Convert to polar form: $r = |z|$, $\theta = \arg z$.
2. Compute r^n and $n\theta$.
3. Reduce $n\theta$ by multiples of 2π (optional, but it simplifies the trigonometry).
4. Convert back: $r^n\cos(n\theta) + i\,r^n\sin(n\theta)$.

Output: z^n in rectangular form.

15.8.2. Worked examples

$(1+i)^{10}$. $r = \sqrt{2}, \theta = \frac{\pi}{4}$. *Try it:* Spiral to (1 + i)¹⁰ = 32i .

$$\begin{aligned}
(1+i)^{10} &= (\sqrt{2})^{10}\left(\cos\frac{10\pi}{4} + i\sin\frac{10\pi}{4}\right) && \text{(De Moivre)}\\
&= 32\left(\cos\frac{5\pi}{2} + i\sin\frac{5\pi}{2}\right) && \left((\sqrt{2})^{10} = 2^5\right)\\
&= 32\left(\cos\frac{\pi}{2} + i\sin\frac{\pi}{2}\right) && \text{(subtract } 2\pi\text{)}\\
&= 32\,(0+i) = 32i.
\end{aligned}$$

Alternative method, as a check: $(1+i)^2 = 1 + 2i + i^2 = 2i$, so $(1+i)^{10} = (2i)^5 = 32\,i^5 = 32i$ ✓. Repeated squaring is a fine alternative when the base squares to something simple; polar form is the general method. Expanding $(1+i)^{10}$ by the binomial theorem would work too, but would take eleven terms.

$(1+\sqrt{3}\,i)^6$. $r = 2, \theta = \frac{\pi}{3}$, so the result is $2^6(\cos 2\pi + i\sin 2\pi) = 64$. A complex number whose sixth power is a positive real: it has rotated through exactly one full turn. *Try it:* See (1 + √3 i)⁶ = 64 .

15.8.3. Application: trigonometric identities for free

Expand $(\cos\theta + i\sin\theta)^3$ in two ways. By De Moivre it is $\cos 3\theta + i\sin 3\theta$. By the binomial expansion $(c+is)^3 = c^3 + 3c^2(is) + 3c(is)^2 + (is)^3$:

$$\begin{aligned}(\cos\theta + i\sin\theta)^3 &= \cos^3\theta + 3i\cos^2\theta\sin\theta - 3\cos\theta\sin^2\theta - i\sin^3\theta \\ &= \big(\cos^3\theta - 3\cos\theta\sin^2\theta\big) + i\big(3\cos^2\theta\sin\theta - \sin^3\theta\big).\end{aligned}$$

Equate real parts, and use $\sin^2\theta = 1 - \cos^2\theta$:

$$\cos 3\theta = \cos^3\theta - 3\cos\theta(1-\cos^2\theta) = 4\cos^3\theta - 3\cos\theta.$$

Equating imaginary parts similarly gives $\sin 3\theta = 3\sin\theta - 4\sin^3\theta$. Multiple-angle formulas need never be memorised again.

15.9. Roots of unity and nth roots

15.9.1. The problem

How many solutions does $z^3 = 8$ have? Over the reals, one: $z = 2$. But polar form lets us run De Moivre backwards. If $z = re^{i\theta}$ then $z^3 = r^3 e^{3i\theta}$, and we need this to equal $8 = 8e^{i\cdot 0}$. Moduli: $r^3 = 8$, so $r = 2$. Arguments: 3θ must point in the same direction as 0 — but "the same direction" means $3\theta = 0, 2\pi, 4\pi, \ldots$. So $\theta = 0, \frac{2\pi}{3}, \frac{4\pi}{3}$ (after that the angles repeat). Three different cube roots: *Try it:* Plot the cube roots of 8 .

$$z = 2, \qquad z = 2e^{2\pi i/3} = -1 + \sqrt{3}\,i, \qquad z = 2e^{4\pi i/3} = -1 - \sqrt{3}\,i.$$

Check one: $(-1+\sqrt{3}i)^2 = 1 - 2\sqrt{3}i - 3 = -2 - 2\sqrt{3}i$, then $(-2-2\sqrt{3}i)(-1+\sqrt{3}i) = 2 - 2\sqrt{3}i + 2\sqrt{3}i - 6i^2 = 2 + 6 = 8$ ✓.



Roots of unity

The **nth roots of unity** are the n complex solutions of $z^n = 1$: $\omega^k = e^{2\pi ik/n}$ for $k = 0, 1, \ldots, n-1$, where $\omega = e^{2\pi i/n}$ is the **primitive** nth root of unity (ω is the Greek letter omega). They lie on the unit circle at the vertices of a regular n-gon, one vertex at 1. For $n \geq 2$ they sum to zero.

Try it: See the sixth roots of unity .

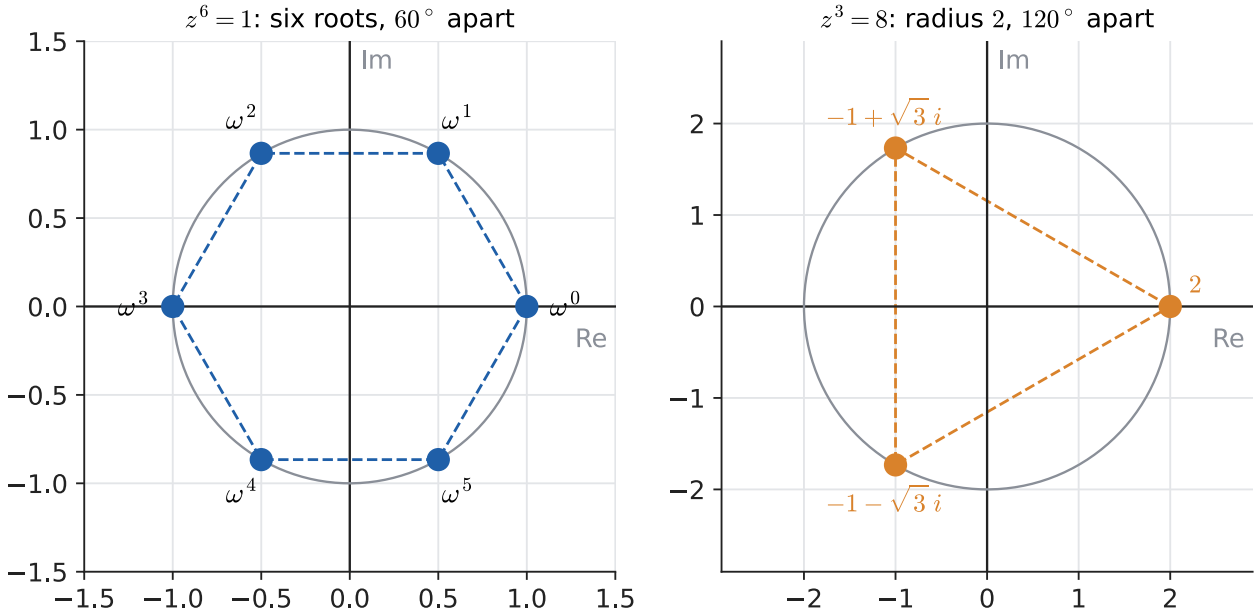


Figure 130. Left: the six sixth roots of unity form a regular hexagon. Right: the three cube roots of 8 form an equilateral triangle of radius 2.

Why do they sum to zero? By symmetry, the regular polygon's vertices balance around the origin. Algebraically, they form a geometric series (Chapter 14) with ratio $\omega \neq 1$: *Try it:* Roots of unity sum to zero .

$$1 + \omega + \omega^2 + \cdots + \omega^{n-1} = \frac{1 - \omega^n}{1 - \omega} = \frac{1 - 1}{1 - \omega} = 0.$$

15.9.2. The method

Example 82. Procedure — all nth roots of a complex number c

Input: nonzero c and a positive integer n.

Steps:

1. Write $c = Re^{i\phi}$ (modulus R, argument ϕ).
2. The roots all have modulus $R^{1/n}$ (the ordinary positive real root).
3. Their arguments are $\dfrac{\phi + 2\pi k}{n}$ for $k = 0, 1, \ldots, n-1$: start at ϕ/n and step by $2\pi/n$.
4. Convert each to rectangular form if needed.

Output: the n distinct nth roots of c, equally spaced on a circle of radius $R^{1/n}$.

```python
import cmath

def nth_roots(c, n):
    """All n complex nth roots of c."""
    r, phi = cmath.polar(c)
    return [cmath.rect(r ** (1 / n), (phi + 2 * cmath.pi * k) / n)
            for k in range(n)]

for z in nth_roots(8, 3):
    print(f"{z:.6f}", f"{z**3:.6f}")
print(sum(nth_roots(1, 6)))
```

```
2.000000+0.000000j 8.000000+0.000000j
-1.000000+1.732051j 8.000000-0.000000j
-1.000000-1.732051j 8.000000-0.000000j
(-1.1102230246251565e-16+4.440892098500626e-16j)
```

The last line is the sum of the sixth roots of unity: zero, up to rounding.

Worked example. Solve $z^4 = -16$. *Try it:* Solve z⁴ = −16 on its circle .

1. $-16 = 16e^{i\pi}$: $R = 16, \phi = \pi$.
2. Modulus of each root: $16^{1/4} = 2$.
3. Arguments: $\frac{\pi + 2\pi k}{4}$ for $k = 0, 1, 2, 3$: $\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}$.
4. Rectangular: $2e^{i\pi/4} = 2\left(\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i\right) = \sqrt{2} + \sqrt{2}i$, and similarly the others: the four roots are $\pm\sqrt{2} \pm \sqrt{2}\,i$ (all four sign combinations).



The principal root is a convention

Calculators and `cmath.sqrt` return one root, the **principal** one (the root whose argument is the principal argument divided by n; for square roots, the one with positive real part, or on the positive imaginary axis). There is no "the" square root of a complex number in the way there is for positive reals; there are always two, negatives of each other. Formulas like $\sqrt{zw} = \sqrt{z}\sqrt{w}$ fail for the same reason the $\sqrt{-1}\sqrt{-1}$ paradox did. In code, if you need all roots, generate them with the procedure above rather than calling `sqrt` or `** (1/n)`.

15.9.3. Where you'd use this

- **Mains electricity (three-phase power):** power stations generate three voltages of the same size, a third of a cycle apart. As phasors they are V_0, $V_0\,\omega^{-1}$ and $V_0\,\omega^{-2}$ with $\omega = e^{2\pi i/3}$: the three cube roots of unity, scaled. Because the cube roots of unity sum to zero, the three voltages add to zero **at every instant**. With a balanced load, the currents returning through the shared neutral wire cancel too, so it can be thin or omitted. That saving in copper is why grids worldwide are three-phase. In the UK each phase is 230 V (325 V peak) at 50 Hz. *Try it:* Watch three-phase phasors cancel .

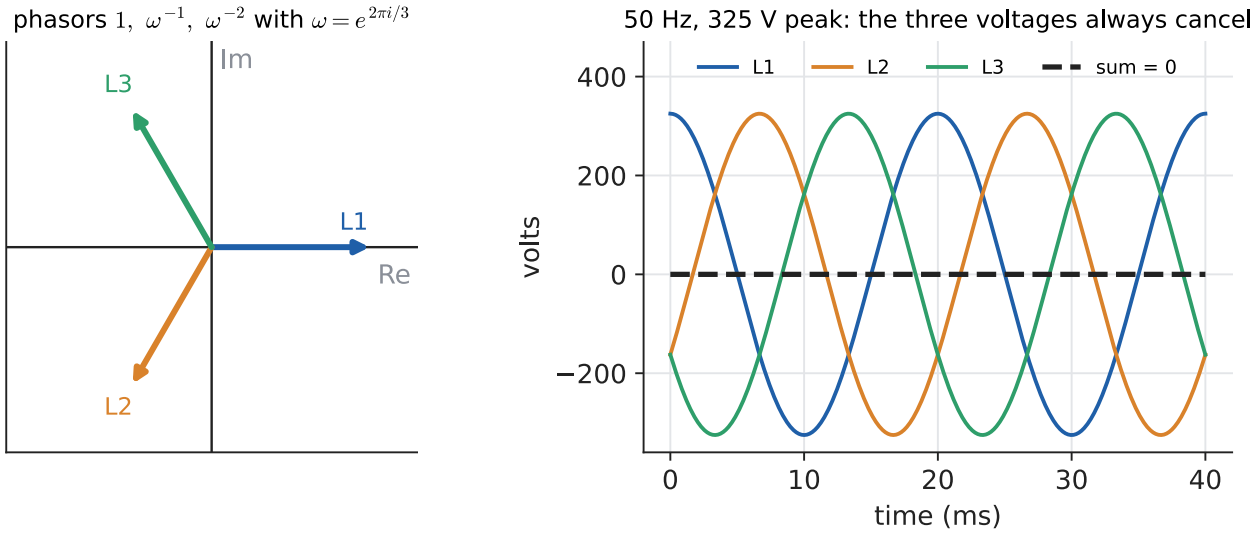


Figure 131. Three-phase mains. Left: the three phasors are the cube roots of unity (scaled), and they sum to zero. Right: the three 50 Hz voltages, and their sum, which is zero at every instant.

- **Graphics:** the vertices of a regular n-gon of radius r centred at c are $c + r\omega^k$ for $k = 0, \dots, n-1$. For a hexagonal map tile of radius 10 at $(50, 50)$, that gives $(60, 50), (55, 58.66), (45, 58.66), (40, 50), (45, 41.34), (55, 41.34)$. Strategy games with hex grids generate their tiles this way, in one line of code.
- **Signal processing:** the FFT below is roots of unity at industrial scale.

Computer-science connection: roots of unity and the FFT

The **discrete Fourier transform** (DFT) takes N samples $x_0, \dots, x_{N-1}$ of a signal and measures how much of each frequency they contain:

$$X_k = \sum_{n=0}^{N-1} x_n \, \omega^{-kn}, \qquad \omega = e^{2\pi i / N}.$$

Each frequency k is tested by multiplying the signal by a "pure tone" ω^{-kn}, a point spinning around the unit circle k times, and adding up. If the signal contains that frequency, the products line up and the sum is large; otherwise they spread around the circle and cancel — exactly the "roots of unity sum to zero" fact.

```python
import cmath, math

def dft(x):
    """Naive discrete Fourier transform: X_k = sum_n x_n * w^(-kn)."""
    N = len(x)
    w = cmath.exp(2j * cmath.pi / N)          # primitive Nth root of unity
    return [sum(x[n] * w ** (-k * n) for n in range(N)) for k in range(N)]

N = 8
signal = [math.cos(2 * math.pi * 2 * n / N) for n in range(N)]  # 2 cycles
X = dft(signal)
print([round(abs(Xk), 6) for Xk in X])
```

```
[0.0, 0.0, 4.0, 0.0, 0.0, 0.0, 4.0, 0.0]
```

The energy appears at frequency 2 (and its mirror image, 6 = 8 − 2). Done naively this costs N^2 multiplications. The **fast Fourier transform** (Cooley and Tukey, 1965) exploits the fact that the Nth roots of unity squared are the $(N/2)$th roots of unity, splitting the problem in half recursively: cost $T(N) = 2T(N/2) + O(N)$, which is $O(N \log N)$ (exactly the recurrence solved in Chapter 14). For $N = 2^{20}$ that is a speed-up of roughly 50,000 times. The FFT underlies MP3 and JPEG compression, Wi-Fi and 4G/5G modulation, MRI imaging, fast multiplication of huge integers, and convolution in signal processing and some neural-network layers. `numpy.fft.fft` gives the same answer as `dft` above, much faster.

Try it: A cosine at fs/4 in the FFT ; DFT arrows line up ; Arrows cancel off-frequency ; Run this in your browser ; Mirror bin k = 6 ; N^2 versus N log N .

15.10. The fundamental theorem of algebra

15.10.1. Every polynomial has a root

We extended the reals to solve one equation, $x^2 + 1 = 0$. A natural worry: will $x^2 = i$ or $x^5 + ix + 3 = 0$ force yet another extension? The answer, and the reason complex numbers are the end of this road, is one of the great theorems of mathematics.



Fundamental theorem of algebra

Fundamental theorem of algebra: every polynomial of degree $n \geq 1$ with complex coefficients has at least one complex root. Consequently it factors completely into n linear factors,

$$p(z) = a_n(z - z_1)(z - z_2)\cdots(z - z_n),$$

so it has exactly n roots in $\mathbb{C}$, **counted with multiplicity** (a repeated factor like $(z-2)^2$ counts twice). A number system in which every polynomial has a root is called **algebraically closed**.

The "consequently" is the factor theorem from Chapter 7: if z_1 is a root, then $(z - z_1)$ is a factor, and the remaining quotient has degree $n - 1$, which again has a root, and so on.

The proofs (Gauss gave several, starting in 1799) need tools beyond this book, but here is the intuition. For a huge circle $|z| = R$, the polynomial behaves like its leading term z^n, which winds around the origin n times as z goes once around the circle. For a tiny circle, $p(z)$ stays close to $p(0)$ and does not wind around the origin at all (if $p(0) \neq 0$). Shrinking the circle continuously from large to small, the image loop must at some point pass through the origin — and that is a root.

15.10.2. Complex roots of real quadratics

For a quadratic with real coefficients and negative discriminant, the quadratic formula (Chapter 8) now simply works:

Worked example. Solve $x^2 - 4x + 13 = 0$. *Try it:* See the roots 2 ± 3i .

$$\begin{aligned} x &= \frac{4 \pm \sqrt{(-4)^2 - 4(1)(13)}}{2(1)} && \text{(quadratic formula, } a=1,\ b=-4,\ c=13\text{)} \\ &= \frac{4 \pm \sqrt{16 - 52}}{2} = \frac{4 \pm \sqrt{-36}}{2} \\ &= \frac{4 \pm 6i}{2} && (\sqrt{-36} = 6i) \\ &= 2 \pm 3i. \end{aligned}$$

Check: $(2+3i)^2 - 4(2+3i) + 13 = (4 + 12i - 9) - 8 - 12i + 13 = 0$ ✓.

Alternative: completing the square. $x^2 - 4x + 13 = (x-2)^2 + 9 = 0$, so $(x-2)^2 = -9$, $x - 2 = \pm 3i$, $x = 2 \pm 3i$. Same answer, and it shows the geometry: the vertex of the parabola is at $(2, 9)$, above the axis; the real part of the roots is the axis of symmetry, and the imaginary part measures how far the parabola "misses" the x-axis.

```python
import cmath

def solve_quadratic(a, b, c):
    """Both roots of ax^2 + bx + c = 0, complex if necessary."""
    disc = b * b - 4 * a * c
    root = cmath.sqrt(disc)          # works for negative disc too
    return (-b + root) / (2 * a), (-b - root) / (2 * a)

print(solve_quadratic(1, -4, 13))
print(solve_quadratic(1, -5, 6))
print(solve_quadratic(1, 0, 1))
```

```
((2+3j), (2-3j))
((3+0j), (2+0j))
(1j, -1j)
```

15.10.3. Conjugate pairs

Notice the roots came as $2 + 3i$ and $2 - 3i$, a conjugate pair. That is no accident. If a polynomial has **real** coefficients and $p(z) = 0$, conjugate everything: since conjugation respects sums and products and leaves real coefficients alone, $p(\bar{z}) = \overline{p(z)} = \bar{0} = 0$. So **non-real roots of real polynomials come in conjugate pairs.** Consequences:

- A real cubic (odd degree) must have at least one real root — non-real roots pair up, and three cannot all pair.
- Multiplying a conjugate pair of factors gives a real quadratic: $(x - (2+3i))(x - (2-3i)) = (x-2)^2 + 9 = x^2 - 4x + 13$. So every real polynomial factors into real linear and real irreducible quadratic factors — a fact used in partial fractions for integration (Volume 2, Chapter 19).

Example. $x^3 - 1 = (x-1)(x^2 + x + 1)$. The quadratic has discriminant $1 - 4 = -3$, so roots $\frac{-1 \pm \sqrt{3}i}{2}$. Together with 1 these are the three cube roots of unity, $e^0, e^{2\pi i/3}, e^{4\pi i/3}$ — the algebra and the geometry agree.

15.10.4. Where you'd use this

- **Car suspension and earthquake-proof buildings:** a car's suspension is a mass on a spring with a damper. Its motion $x(t)$ satisfies a differential equation whose behaviour is set by the roots of a quadratic, its **characteristic equation** (deriving it needs the derivatives of Volume 2, Chapter 17 and a little more). For a typical setting, that quadratic might be $r^2 + 2r + 26 = 0$, with roots $r = -1 \pm 5i$. The real part -1 says the bounce dies away like e^{-t}. The imaginary part 5 says it

oscillates at 5 rad/s, a period of $2\pi/5 \approx 1.26$ s. If the damper wears out, the real part moves towards 0 and the car keeps bouncing. Engineers tune the damper so the roots sit where the ride is comfortable. *Try it:* Plot the suspension roots .

- **Control and computing:** the same rule decides whether an autopilot, a drone's stabiliser or a digital audio filter is stable. Find the roots (the **poles**) of its characteristic polynomial and check they are on the safe side: negative real part for continuous systems, inside the unit circle for digital ones. `numpy.roots` and the conjugate-pair rule do the work.

Choosing a method for polynomial equations

- **Quadratic:** quadratic formula (or completing the square); complex roots come straight out.
- $z^n = c$: polar form and the nth-roots procedure; never expand.
- **Real polynomial with a known complex root:** its conjugate is a root too; divide out the real quadratic factor.
- **General higher degree:** there is no formula using radicals for degree 5 and above (Abel and Galois, 1820s). Use numerical root-finding (Volume 2, Chapter 32); `numpy.roots` computes the eigenvalues of a "companion matrix" (Volume 2, Chapter 28).

US curriculum

Complex numbers are introduced in **Algebra II** (Common Core HSN-CN.A: the imaginary unit and arithmetic, HSN-CN.C.7: complex solutions of quadratics, HSN-CN.C.9: the fundamental theorem of algebra, stated). The complex plane, polar form, De Moivre's theorem, and roots appear in **Precalculus** (HSN-CN.B, a "+" standard often reserved for advanced tracks). Euler's formula is usually not met until college, though strong students see it in **AP Calculus BC** through Taylor series.

15.11. Applications

15.11.1. Electrical engineering: AC circuits

In alternating-current circuits, voltages and currents oscillate sinusoidally, $V(t) = V_0 \cos(\omega t + \phi)$ (here ω is the angular frequency, not a root of unity). Resistors, capacitors, and inductors each respond differently: a resistor's current is in step with the voltage; an inductor's current lags by a quarter cycle; a capacitor's leads by a quarter cycle. Tracking these phase shifts with trigonometric identities is miserable.

Engineers instead represent each oscillation by a complex number, its **phasor** $V_0 e^{i\phi}$ (the oscillation is the real part of $V_0 e^{i\phi} e^{i\omega t}$). A "quarter-cycle shift" is multiplication by $\pm i$ — the quarter-turn again. Each component gets a complex **impedance** (generalised resistance):

$$Z_R = R, \qquad Z_L = i\omega L, \qquad Z_C = \frac{1}{i\omega C} = -\frac{i}{\omega C},$$

and Ohm's law $V = IZ$ works with complex numbers. Components in series add their impedances; in parallel, the reciprocals add — exactly as for plain resistors.

Worked example. A 30 Ω resistor in series with an inductor of reactance $\omega L = 40$ Ω is driven by a voltage of amplitude 100 V. Then $Z = 30 + 40i$ Ω, and *Try it:* Solve the AC circuit with phasors .

$$\begin{aligned} I = \frac{V}{Z} &= \frac{100}{30 + 40i} = \frac{100(30 - 40i)}{30^2 + 40^2} \qquad \text{(multiply by the conjugate)} \\ &= \frac{3000 - 4000i}{2500} = 1.2 - 1.6i. \end{aligned}$$

The current's amplitude is $|I| = \sqrt{1.44 + 2.56} = 2$ A (equivalently $100/|Z| = 100/50$), and its phase is $\arg I = -\arctan \frac{1.6}{1.2} \approx -53.1°$: it lags the voltage by $53.1°$. One complex division replaced a page of trigonometry.

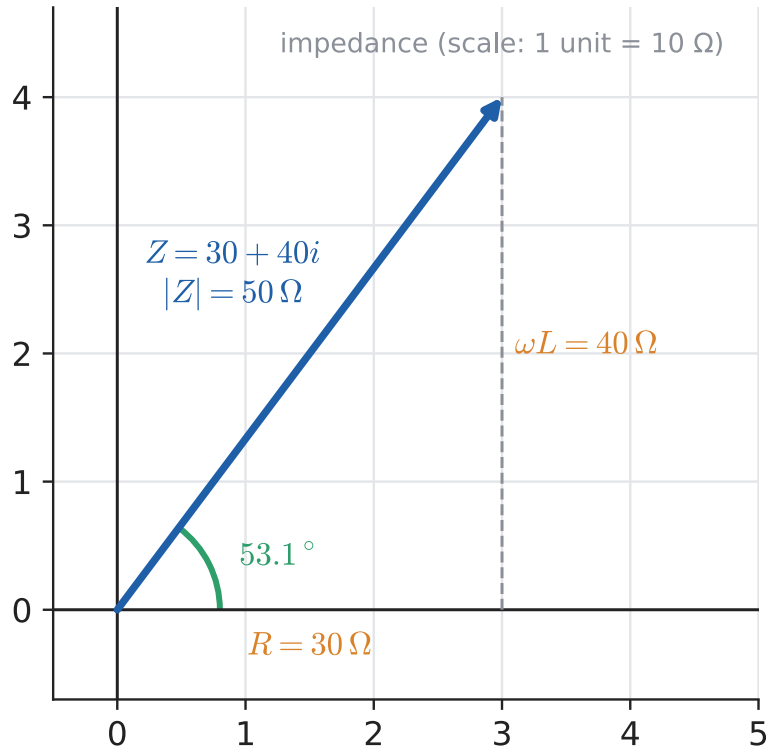


Figure 132. The impedance of a resistor and inductor in series. Its modulus gives the effective resistance; its argument gives the phase lag.

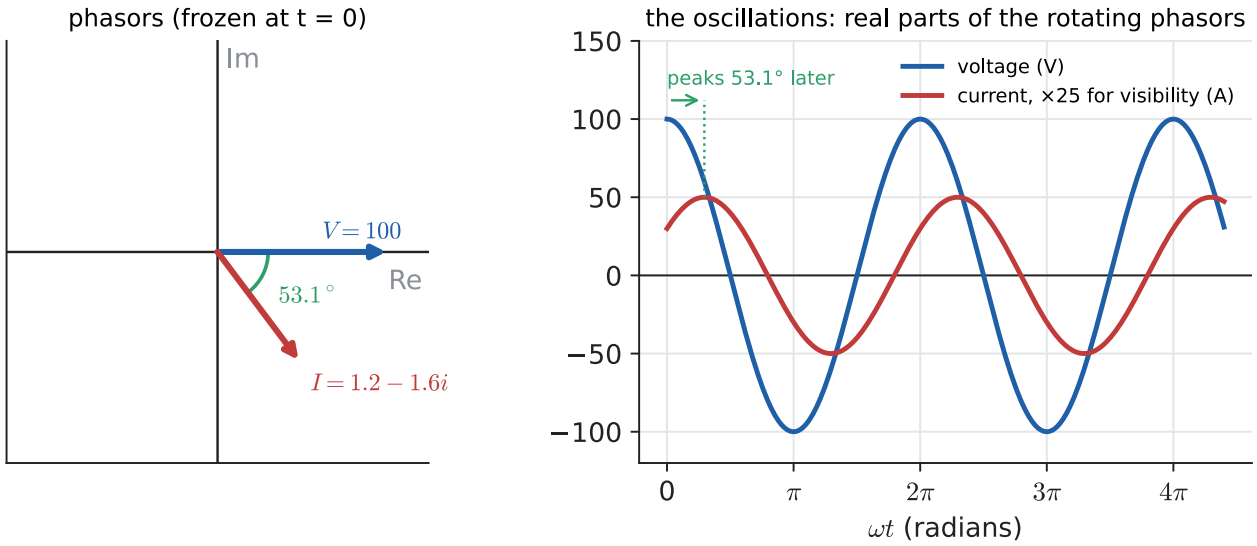


Figure 133. The same example in time. Left: the voltage and current phasors at $t = 0$; the current lags by 53.1°. Right: the actual oscillations are the real parts of these phasors as they rotate together at angular frequency ω, so the current's peaks arrive 53.1° (about 0.15 of a cycle) after the voltage's.

15.11.2. Signal processing

Any reasonable periodic signal is a sum of sines and cosines (**Fourier series**), and Euler's formula packs each cosine–sine pair into one complex exponential: $\cos \omega t = \frac{e^{i\omega t} + e^{-i\omega t}}{2}$. Filtering a signal (removing hiss, isolating a radio station, equalising audio) becomes multiplying each frequency component by a complex number that scales its amplitude and shifts its phase. The DFT and FFT from the sidebar above are the computational form.

15.11.3. Quantum mechanics (an aside)

In quantum mechanics, complex numbers are not a convenience but part of the foundations. The state of a system is described by complex **amplitudes**; the probability of an outcome is the squared modulus $|\psi|^2$ of its amplitude. Because amplitudes have phases, two paths to the same outcome can cancel (destructive interference) or reinforce, which is what the double-slit experiment shows. Schrödinger's equation, $i\hbar\frac{\partial\psi}{\partial t} = \hat{H}\psi$, has i right at the front. Quantum computing algorithms (including Shor's factoring algorithm, which uses a quantum Fourier transform built from roots of unity) are manipulations of these complex amplitudes.

15.11.4. Elsewhere

- **Fluid flow and aerodynamics:** complex functions map simple flows (around a cylinder) to complicated ones (around an aerofoil) — the Joukowski transform.
- **Control engineering:** a system is stable when the roots of its characteristic polynomial (often complex) have negative real parts.
- **Linear algebra:** a rotation matrix has no real eigenvectors (nothing stays on its own line when you rotate), but it has complex eigenvalues $e^{\pm i\theta}$ (Volume 2, Chapter 28).

15.12. The Mandelbrot set

The most famous picture in mathematics comes from iterating one tiny complex formula.

15.12.1. The problem

Pick a complex number c. Start with $z_0 = 0$ and repeatedly apply $z_{n+1} = z_n^2 + c$ (a recurrence, Chapter 14). Does the sequence stay bounded, or does it run off to infinity? The **Mandelbrot set** is the set of all c for which it stays bounded.

Try a few by hand: *Try it:* Follow the orbit of c = i.

- $c = 1$: $0, 1, 2, 5, 26, 677, \ldots$ — escapes. Not in the set.
- $c = -1$: $0, -1, 0, -1, 0, \ldots$ — a cycle. In the set.
- $c = i$: $0,\ i,\ i^2 + i = -1 + i,\ (-1+i)^2 + i = -2i + i = -i,\ (-i)^2 + i = -1 + i, \ldots$ — cycles between $-1 + i$ and $-i$. In the set.

A useful fact makes computation possible: **once $|z_n| > 2$, the sequence is guaranteed to escape to infinity.** (Roughly: when $|c| \le 2$ and $|z| > 2$, squaring makes $|z|$ grow faster than adding c can pull it back; and if $|c| > 2$ the very first step already exceeds 2.) So we iterate until $|z| > 2$ or until we give up after some maximum number of steps.

Example 83. Procedure — escape-time test for the Mandelbrot set

Input: a complex number c; an iteration limit N (say 50–1000).

Steps:

1. Set $z \leftarrow 0$.
2. For $n = 0, 1, \ldots, N-1$:
   a. Update $z \leftarrow z^2 + c$.
   b. If $|z| > 2$, stop: c is **outside** the set; report the escape step n.
3. If the loop finishes without escaping, report c as (probably) **inside** the set.

Output: inside / outside, and for outside points the escape step, which is used to choose a colour.

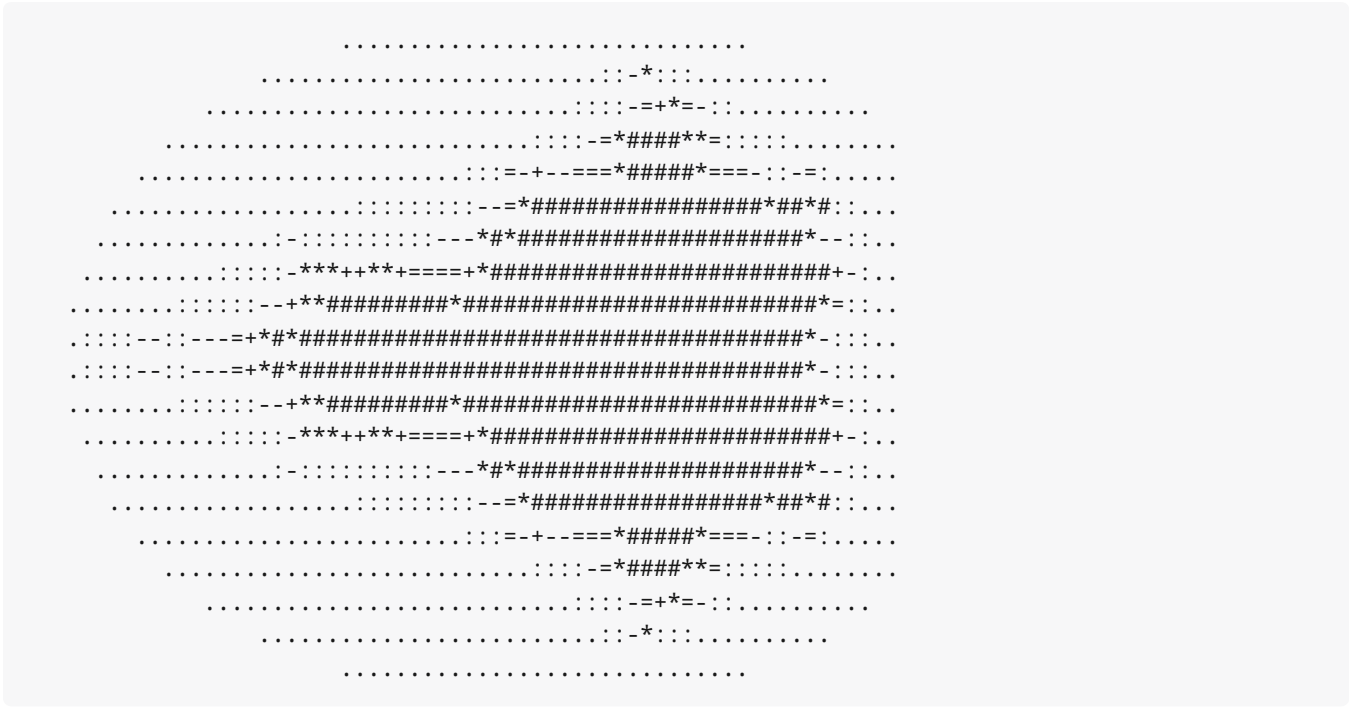
To draw the set, run this for the c at the centre of every pixel.

PYTHON

```
def escape_count(c, max_iter=50):
    """Iterate z -> z^2 + c from z = 0; return the step at which |z| > 2,
    or max_iter if it never escapes (c is then taken to be in the set)."""
    z = 0j
    for n in range(max_iter):
        z = z * z + c
        if abs(z) > 2:
            return n
    return max_iter

def mandelbrot_ascii(width=64, height=20, max_iter=50):
    shades = " .:-=+*"              # outside: escapes fast ... escapes slowly
    lines = []
    for row in range(height):
        y = 1.2 - 2.4 * row / (height - 1)
        line = ""
        for col in range(width):
            x = -2.1 + 2.7 * col / (width - 1)
            n = escape_count(complex(x, y), max_iter)
            if n == max_iter:
                line += "#"             # inside the set
            else:
                line += shades[min(n // 2, len(shades) - 1)]
        lines.append(line)
    return "\n".join(lines)

print(mandelbrot_ascii())
```



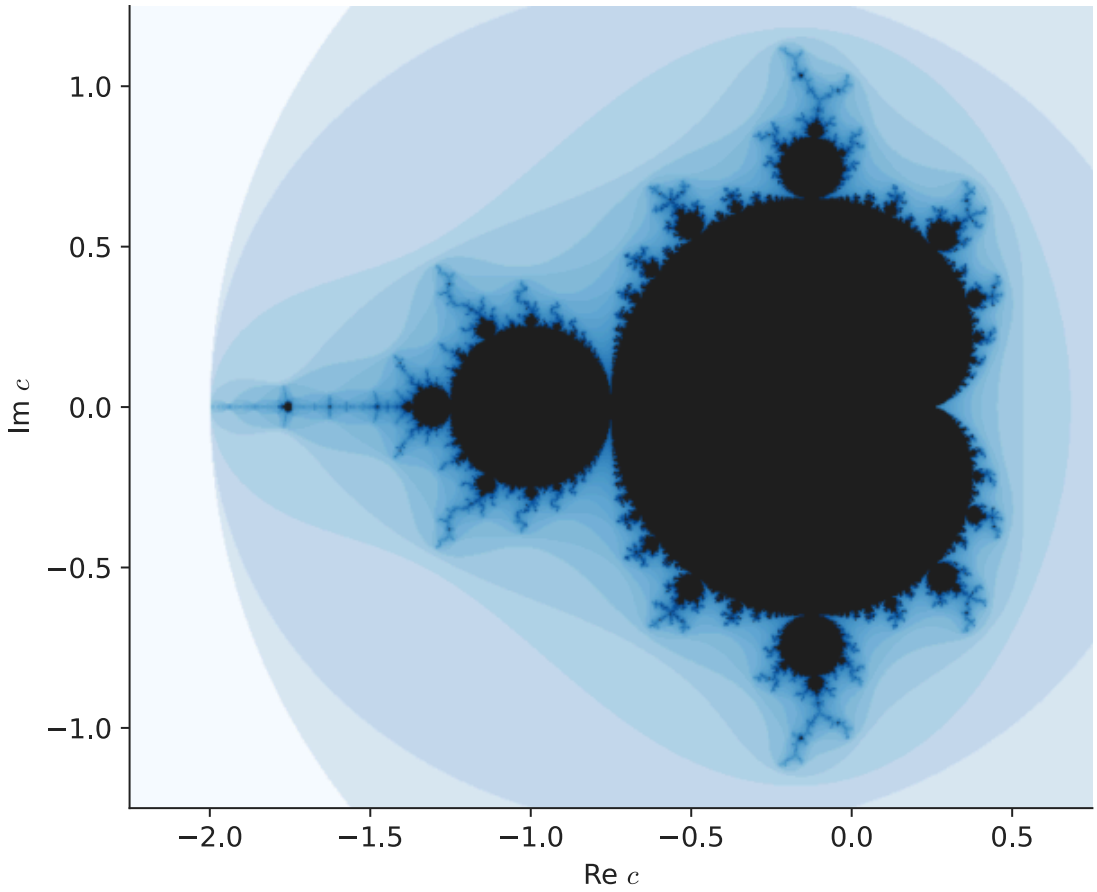


Figure 134. The Mandelbrot set (black), with points outside shaded by how quickly they escape. The boundary is infinitely intricate.



Where the failure modes hide

The escape-time test can only ever prove that a point is *outside*. A point that has not escaped after N steps might escape at step $N+1$; near the boundary, points can take millions of iterations. Raising N sharpens the picture but costs time, and deep zooms eventually run out of double-precision accuracy (pixels closer together than about 10^{-16} times the coordinates are indistinguishable), which is why deep-zoom renderers use arbitrary precision or clever perturbation methods (Volume 2, Chapter 32).

Try it: Zoom past double precision .



Iteration, chaos, and beauty

Benoit Mandelbrot produced the first crude printouts of the set at IBM in 1980. Its boundary is a **fractal**: zoom in anywhere and new detail appears forever, including tiny distorted copies of the whole set. Its area is still not known exactly (about 1.506). Mandelbrot coined the word **fractal** from the Latin *fractus*, broken. A deep theorem says the set is connected — all one piece — even though the thin filaments make it look like scattered islands in any finite picture.

Explore the Mandelbrot set

Click any point c in the plane and watch the orbit $0, c, c^2 + c, \ldots$ drawn as a path of connected dots: spiralling into a fixed point, cycling, or flying off. Zoom into the boundary with a scroll wheel and a slider for the iteration limit, and see where double precision gives out.

Try it: Explore the Mandelbrot set

Multiply and watch

Draw a shape (a letter F works well, since it has no symmetry). Drag a complex number w around the plane and watch the shape multiplied by w update live: rotating as w moves around a circle, growing and shrinking as w moves in and out, and turning exactly a quarter-turn when $w = i$. A second mode shows $z \mapsto z^2$ distorting a grid, doubling all angles.

Try it: Multiply and watch

15.13. Interactive tools

Browser tools on the companion website that go with this chapter. Each example link in the chapter opens one of them with that example already set up.

- Complex numbers, visually
- Unit circle and waves
- Graphing calculator
- Fourier series
- Spectrum analyser
- Language sandbox

15.14. Exercises

Recognition

1. Give the real and imaginary parts of: (a) $7 - 2i$ (b) $5i$ (c) -3 (d) $\frac{1+i}{2}$.
2. Simplify i^{2026}, i^{-1}, and i^{4k+3} for an integer k.
3. Describe, in words, the effect on the complex plane of multiplying every point by: (a) 3 (b) -1 (c) i (d) $-i$ (e) $1 + i$.

Application

4. Compute in the form $a + bi$: (a) $(2 + 3i)(4 - i)$ (b) $(3 - 2i)^2$ (c) $\dfrac{5 - i}{2 + 3i}$.

5. Find the modulus and principal argument of (a) $-2+2i$ (b) $-\sqrt{3}-i$ (c) -4.
6. Use De Moivre's theorem to compute $(-1+i)^8$. Check by repeated squaring.
7. Solve (a) $x^2+6x+13=0$ (b) $z^4=-16$ (c) $z^3=-8i$.
8. Rotate the point $(4,1)$ by 90° anticlockwise about the point $(1,1)$ using complex multiplication.

Problem solving

9. Show that $|z|^2=z\bar{z}$ and use it to prove $|zw|=|z||w|$ without polar form.
10. Use De Moivre's theorem to express $\cos 4\theta$ in terms of $\cos\theta$.
11. A cubic with real coefficients has roots 3 and $1-2i$. Find its third root and write the cubic (leading coefficient 1) with real coefficients.
12. Decide by iterating whether $c=0.5$ and $c=-i$ are in the Mandelbrot set.

▼ **Answers**

1. (a) Re = 7, Im = -2. (b) Re = 0, Im = 5. (c) Re = -3, Im = 0. (d) Re = $\frac{1}{2}$, Im = $\frac{1}{2}$.
2. $2026=4\cdot 506+2$, so $i^{2026}=i^2=-1$. $i^{-1}=\frac{1}{i}=\frac{i}{i^2}=-i$. $i^{4k+3}=(i^4)^k i^3=-i$.
3. (a) Stretch away from 0 by a factor 3. (b) Rotate 180° (point reflection through 0). (c) Rotate 90° anticlockwise. (d) Rotate 90° clockwise. (e) $|1+i|=\sqrt{2}$, $\arg=45^\circ$: rotate 45° anticlockwise and scale by $\sqrt{2}$.
4. (a) $8-2i+12i-3i^2=11+10i$. (b) $9-12i+4i^2=5-12i$. (c) $\frac{(5-i)(2-3i)}{4+9}=\frac{10-15i-2i+3i^2}{13}=\frac{7-17i}{13}=\frac{7}{13}-\frac{17}{13}i$.
5. (a) $|z|=\sqrt{8}=2\sqrt{2}$; second quadrant, reference angle $\frac{\pi}{4}$, so $\arg=\frac{3\pi}{4}$ (135°). (b) $|z|=\sqrt{3+1}=2$; third quadrant, reference angle $\arctan\frac{1}{\sqrt{3}}=\frac{\pi}{6}$, so $\arg=-(\pi-\frac{\pi}{6})=-\frac{5\pi}{6}$ (-150°). (c) $|z|=4$, $\arg=\pi$.
6. $-1+i=\sqrt{2}\,e^{3\pi i/4}$. $(-1+i)^8=(\sqrt{2})^8 e^{6\pi i}=16\,e^{0}=16$. Check: $(-1+i)^2=1-2i-1=-2i$; $(-2i)^2=-4$; $(-4)^2=16$ ✓.
7. (a) $x=\frac{-6\pm\sqrt{36-52}}{2}=\frac{-6\pm 4i}{2}=-3\pm 2i$. (b) $\pm\sqrt{2}\pm\sqrt{2}\,i$ (worked in the text). (c) $-8i=8e^{-i\pi/2}$: modulus of roots 2, arguments $\frac{-\pi/2+2\pi k}{3}=-\frac{\pi}{6},\frac{\pi}{2},\frac{7\pi}{6}$. Roots: $2e^{-i\pi/6}=\sqrt{3}-i$, $2e^{i\pi/2}=2i$, $2e^{7i\pi/6}=-\sqrt{3}-i$. Check: $(2i)^3=8i^3=-8i$ ✓.
8. $p=4+i$, $c=1+i$. $p-c=3$; times i: $3i$; plus c: $1+4i$. The image is $(1,4)$.
9. With $z=a+bi$: $z\bar{z}=a^2-(bi)^2=a^2+b^2=|z|^2$. Then $|zw|^2=(zw)\overline{(zw)}=zw\,\bar{z}\,\bar{w}=(z\bar{z})(w\bar{w})=|z|^2|w|^2$; take non-negative square roots.
10. Real part of $(c+is)^4=c^4+4c^3(is)+6c^2(is)^2+4c(is)^3+(is)^4$ is $c^4-6c^2s^2+s^4$. Substitute $s^2=1-c^2$: $c^4-6c^2(1-c^2)+(1-c^2)^2=c^4-6c^2+6c^4+1-2c^2+c^4=8c^4-8c^2+1$. So $\cos 4\theta=8\cos^4\theta-8\cos^2\theta+1$. (Check $\theta=0$: $8-8+1=1$ ✓.)

11. Real coefficients, so $1+2i$ is also a root.
$(x-(1-2i))(x-(1+2i)) = (x-1)^2+4 = x^2-2x+5$. The cubic is
$(x-3)(x^2-2x+5) = x^3-5x^2+11x-15$.

12. $c=0.5$: $0, 0.5, 0.75, 1.0625, 1.6289, 3.1533$ — exceeds 2, escapes: not in the set. $c=-i$:
$0, -i, (-i)^2-i = -1-i, (-1-i)^2-i = 2i-i = i, i^2-i = -1-i, \ldots$ — cycles between $-1-i$ and i: bounded, in the set.

15.15. Mastery check

1. Compute $(3-2i)^2$ and its modulus.
2. Write $\dfrac{4+2i}{1-i}$ in the form $a+bi$.
3. Write $1-\sqrt{3}\,i$ in polar (exponential) form, then compute $(1-\sqrt{3}\,i)^5$ in rectangular form.
4. Find all three cube roots of $-8i$ and sketch them.
5. Solve $x^2-2x+10=0$ and check one root by substitution.
6. Explain, in a sentence, why multiplying by i twice is the same as multiplying by -1, in terms of geometry.
7. Use Euler's formula to write $e^{i\pi/3}$ in the form $a+bi$, and explain why $|e^{i\theta}|=1$ for every real θ.

Decision rule: All correct without hesitation → move on to Volume 2, Chapter 16 . Stuck on 1–2 → reread Section 15.2 and Section 15.4. Stuck on 3–4 → reread Section 15.5, Section 15.6, and Section 15.9. Stuck on 6–7 → revisit Section 15.5 and Section 15.7 with the diagrams, then the exercises.

▼ **Mastery check answers**

1. $(3-2i)^2 = 9-12i+4i^2 = 5-12i$; $|5-12i| = \sqrt{25+144} = 13$ (equivalently $|3-2i|^2 = 13$).
2. $\frac{(4+2i)(1+i)}{(1-i)(1+i)} = \frac{4+4i+2i+2i^2}{2} = \frac{2+6i}{2} = 1+3i$. Check:
$(1+3i)(1-i) = 1-i+3i-3i^2 = 4+2i$ ✓.
3. $|1-\sqrt{3}i| = 2$; fourth quadrant, reference angle $\frac{\pi}{3}$, so $1-\sqrt{3}i = 2e^{-i\pi/3}$. Then
$(2e^{-i\pi/3})^5 = 32e^{-5i\pi/3} = 32e^{i\pi/3}$ (add 2π) $= 32\left(\frac{1}{2}+\frac{\sqrt{3}}{2}i\right) = 16+16\sqrt{3}\,i$.
4. $-8i = 8e^{-i\pi/2}$; roots $\sqrt{3}-i$, $2i$, $-\sqrt{3}-i$, on a circle of radius 2, $120°$ apart (an equilateral triangle with a vertex at $2i$).
5. $x = \frac{2\pm\sqrt{4-40}}{2} = \frac{2\pm 6i}{2} = 1\pm 3i$. Check $1+3i$:
$(1+3i)^2-2(1+3i)+10 = (1+6i-9)-2-6i+10 = 0$ ✓.

6. Multiplying by i rotates the plane a quarter-turn about 0; two quarter-turns make a half-turn, which sends every point z to $-z$ — the same as multiplying by -1. That is what $i^2 = -1$ means.
7. $e^{i\pi/3} = \cos\frac{\pi}{3} + i\sin\frac{\pi}{3} = \frac{1}{2} + \frac{\sqrt{3}}{2}i$. $|e^{i\theta}| = \sqrt{\cos^2\theta + \sin^2\theta} = 1$ by the Pythagorean identity: every $e^{i\theta}$ lies on the unit circle.

Appendix A: Cheat Sheet

This is the "I remember learning this — remind me what it means" appendix. Every row is a reminder, not a lesson: if a formula looks unfamiliar, follow the chapter link and read the explanation in context. Definitions of terms live in the Glossary (Appendix B); symbols and how to read them aloud are in Appendix C.

Conventions used throughout: letters $a, b, c, \ldots$ are real numbers unless stated otherwise; denominators are assumed nonzero; logarithms and square roots are applied only where they are defined; $\ln$ is the natural logarithm (base e); angles in calculus formulas are in radians; C is an arbitrary constant of integration.

A.1. Arithmetic and fractions

Rule	Formula or statement	Ch.
Order of operations	Brackets, then powers, then $\times$/$\div$ left to right, then $+$/$-$ left to right. $2 + 3 \times 4^2 = 2 + 48 = 50$	Chapter 1
Signs	$(-a)(-b) = ab$, $(-a)b = -ab$, $a - (-b) = a + b$	Chapter 1
Division with remainder	$a = qb + r$ with $0 \le r < b$ (quotient q, remainder r)	Chapter 1
Absolute value	$\lvert x \rvert = x$ if $x \ge 0$, $-x$ if $x < 0$; distance between a and b is $\lvert a - b \rvert$	Chapter 1
Place value in base b	$(d_k \dots d_1 d_0)_b = \sum_{i=0}^{k} d_i \, b^i$, e.g. $1011_2 = 8 + 0 + 2 + 1 = 11$	Chapter 1
gcd and lcm	$\gcd(a,b) \cdot \operatorname{lcm}(a,b) = ab$ for positive integers	Chapter 1
Divisibility tests	By 3 or 9: digit sum divisible by 3 or 9. By 11: alternating digit sum divisible by 11. By 4: last two digits divisible by 4.	Chapter 1
Equivalent fractions	$\frac{a}{b} = \frac{ka}{kb}$ for $k \neq 0$	Chapter 2
Add / subtract fractions	$\frac{a}{b} \pm \frac{c}{d} = \frac{ad \pm bc}{bd}$ (or use the least common denominator)	Chapter 2
Multiply fractions	$\frac{a}{b} \cdot \frac{c}{d} = \frac{ac}{bd}$	Chapter 2
Divide fractions	$\frac{a}{b} \div \frac{c}{d} = \frac{a}{b} \cdot \frac{d}{c} = \frac{ad}{bc}$	Chapter 2
Percent of a quantity	$p\%$ of x is $\frac{p}{100}\,x$	Chapter 2
Percentage change	$\frac{\text{new} - \text{old}}{\text{old}} \times 100\%$	Chapter 2
Successive changes	Multiply factors: up 10% then down 10% is $1.1 \times 0.9 = 0.99$, a 1% fall overall	Chapter 2

Rule	Formula or statement	Ch.
Ratio split	Sharing T in the ratio $a : b$ gives $\frac{a}{a+b}T$ and $\frac{b}{a+b}T$	Chapter 2
Repeating decimal to fraction	$0.\overline{d_1 \ldots d_k} = \frac{d_1 \ldots d_k}{10^k - 1}$, e.g. $0.\overline{27} = \frac{27}{99} = \frac{3}{11}$	Chapter 2

A.2. Exponents, roots, and logarithms

Rule	Formula	Ch.
Product of powers	$a^m a^n = a^{m+n}$	Chapter 3
Quotient of powers	$\dfrac{a^m}{a^n} = a^{m-n}$	Chapter 3
Power of a power	$(a^m)^n = a^{mn}$	Chapter 3
Power of a product / quotient	$(ab)^n = a^n b^n$, $\left(\dfrac{a}{b}\right)^n = \dfrac{a^n}{b^n}$	Chapter 3
Zero and negative exponents	$a^0 = 1$ $(a \neq 0)$, $a^{-n} = \dfrac{1}{a^n}$	Chapter 3
Fractional exponents	$a^{1/n} = \sqrt[n]{a}$, $a^{m/n} = \left(\sqrt[n]{a}\right)^m = \sqrt[n]{a^m}$ $(a > 0)$	Chapter 3
Root rules	$\sqrt{ab} = \sqrt{a}\sqrt{b}$, $\sqrt{a/b} = \sqrt{a}/\sqrt{b}$ for $a, b \geq 0$; but $\sqrt{a+b} \neq \sqrt{a} + \sqrt{b}$	Chapter 3
Square root of a square	$\sqrt{x^2} = \lvert x \rvert$	Chapter 3
Rationalising	$\dfrac{1}{\sqrt{a}} = \dfrac{\sqrt{a}}{a}$, $\dfrac{1}{\sqrt{a}+\sqrt{b}} = \dfrac{\sqrt{a}-\sqrt{b}}{a-b}$	Chapter 3
Scientific notation	$x = m \times 10^n$ with $1 \leq \lvert m \rvert < 10$; multiply by multiplying mantissas and adding exponents	Chapter 3
Definition of logarithm	$\log_b x = y \iff b^y = x$ $(b > 0, b \neq 1, x > 0)$	Chapter 12
Product / quotient laws	$\log_b(xy) = \log_b x + \log_b y$, $\log_b(x/y) = \log_b x - \log_b y$	Chapter 12
Power law	$\log_b(x^k) = k \log_b x$	Chapter 12
Special values	$\log_b 1 = 0$, $\log_b b = 1$, $b^{\log_b x} = x$, $\log_b(b^x) = x$	Chapter 12

Rule	Formula	Ch.
Change of base	$\log_b x = \dfrac{\ln x}{\ln b} = \dfrac{\log_{10} x}{\log_{10} b}$; e.g. $\log_2 x \approx 3.32 \log_{10} x$	Chapter 12
Exponential via e	$a^x = e^{x \ln a}$	Chapter 12
Not a law	$\log(x+y) \neq \log x + \log y$; $(\log x)^2 \neq \log(x^2)$	Chapter 12
Exponential growth / decay	$N(t) = N_0 e^{kt}$ (growth $k>0$, decay $k<0$); half-life form $N_0\left(\frac{1}{2}\right)^{t/T}$	Chapter 12
Doubling time	$T_2 = \dfrac{\ln 2}{k}$; for $r\%$ per period, roughly $70/r$ periods ("rule of 70")	Chapter 12
Compound interest	$A = P\left(1 + \dfrac{r}{n}\right)^{nt}$; continuously compounded $A = Pe^{rt}$	Chapter 12
Defining e	$e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n \approx 2.71828$	Chapter 12

A.3. Algebraic identities and polynomials

Identity	Formula	Ch.
Distributive law	$a(b+c) = ab + ac$	Chapter 4
Expanding two brackets	$(a+b)(c+d) = ac + ad + bc + bd$	Chapter 4
Square of a sum / difference	$(a \pm b)^2 = a^2 \pm 2ab + b^2$	Chapter 4
Difference of two squares	$a^2 - b^2 = (a+b)(a-b)$	Chapter 4
Cube of a sum / difference	$(a \pm b)^3 = a^3 \pm 3a^2b + 3ab^2 \pm b^3$	Chapter 7
Sum / difference of cubes	$a^3 + b^3 = (a+b)(a^2 - ab + b^2)$, $a^3 - b^3 = (a-b)(a^2 + ab + b^2)$	Chapter 7
Difference of nth powers	$a^n - b^n = (a-b)(a^{n-1} + a^{n-2}b + \cdots + ab^{n-2} + b^{n-1})$	Chapter 7
Binomial theorem	$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k}b^k$	Volume 3, Chapter 24
Remainder theorem	Dividing $p(x)$ by $(x-r)$ leaves remainder $p(r)$	Chapter 7
Factor theorem	$(x-r)$ is a factor of $p(x) \iff p(r) = 0$	Chapter 7
Rational root theorem	If p/q (lowest terms) is a root of $a_n x^n + \cdots + a_0$ with integer coefficients, then p divides a_0 and q divides a_n	Chapter 7
Number of roots	A degree-n polynomial has at most n real roots, and exactly n complex roots counted with multiplicity	Chapter 7
End behaviour	Determined by the leading term $a_n x^n$: its sign and whether n is even or odd	Chapter 7

Identity	Formula	Ch.
Simplifying rational expressions	Cancel common *factors*, never common *terms*: $\frac{x(x+1)}{x} = x+1$ for $x \neq 0$, but $\frac{x+1}{x} \neq 1$	Chapter 4

A.4. Quadratics

Fact	Formula	Ch.
Quadratic formula	$ax^2+bx+c=0 \implies x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$ ($a \neq 0$)	Chapter 8
Discriminant	$\Delta = b^2-4ac$: $\Delta > 0$ two real roots; $\Delta = 0$ one repeated real root; $\Delta < 0$ two complex-conjugate roots	Chapter 8
Completing the square	$x^2+bx = \left(x+\frac{b}{2}\right)^2 - \frac{b^2}{4}$	Chapter 8
Vertex form	$ax^2+bx+c = a(x-h)^2+k$ with $h = -\frac{b}{2a}$, $k = c - \frac{b^2}{4a}$; vertex (h,k), axis of symmetry $x=h$	Chapter 8
Factored form	$a(x-r_1)(x-r_2)$ where r_1, r_2 are the roots	Chapter 8
Sum and product of roots (Vieta)	$r_1+r_2 = -\frac{b}{a}$, $r_1 r_2 = \frac{c}{a}$	Chapter 8
Numerically stable roots	$q = -\frac{1}{2}\left(b + \operatorname{sign}(b)\sqrt{\Delta}\right)$, then $x_1 = q/a$, $x_2 = c/q$ (avoids cancellation when $b^2 \gg 4ac$)	Chapter 8

A.5. Linear equations, lines, and coordinates

Fact	Formula	Ch.
Inequalities	Multiplying or dividing both sides by a negative number reverses the inequality: $-2x < 6 \iff x > -3$	Chapter 5
Absolute-value inequality	$\lvert x - a \rvert < r \iff a - r < x < a + r$	Chapter 5
Slope	$m = \dfrac{y_2 - y_1}{x_2 - x_1} = \dfrac{\text{rise}}{\text{run}}$	Chapter 5
Slope-intercept form	$y = mx + b$ (slope m, y-intercept b)	Chapter 5
Point-slope form	$y - y_1 = m(x - x_1)$	Chapter 5
Standard form	$Ax + By = C$; slope $-A/B$	Chapter 5
Parallel / perpendicular	Parallel: $m_1 = m_2$. Perpendicular: $m_1 m_2 = -1$	Chapter 5
Two equations, two unknowns (Cramer)	$ax + by = e,\ cx + dy = f \implies x = \dfrac{ed - bf}{ad - bc},\ y = \dfrac{af - ec}{ad - bc}$ (if $ad - bc \neq 0$)	Chapter 6
Number of solutions of a linear system	Exactly one, none (inconsistent), or infinitely many; never exactly two	Chapter 6
Distance	$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$	Chapter 10
Midpoint	$\left(\dfrac{x_1 + x_2}{2}, \dfrac{y_1 + y_2}{2} \right)$	Chapter 10
Circle	Centre (h, k), radius r: $(x - h)^2 + (y - k)^2 = r^2$	Chapter 10
Point-to-line distance	From (x_0, y_0) to $Ax + By + C = 0$: $\dfrac{\lvert Ax_0 + By_0 + C \rvert}{\sqrt{A^2 + B^2}}$	Chapter 10

Fact	Formula	Ch.
Transformations of graphs	$y = f(x - h) + k$ shifts right h, up k; $y = af(x)$ stretches vertically by a; $y = f(bx)$ compresses horizontally by b; $y = -f(x)$ reflects in the x-axis; $y = f(-x)$ in the y-axis	Chapter 11
Composition and inverse	$(f \circ g)(x) = f(g(x))$; $f^{-1}(f(x)) = x$; graph of f^{-1} is the reflection of the graph of f in $y = x$	Chapter 11

A.6. Geometry

Shape or fact	Formula	Ch.
Pythagoras	Right triangle with legs a, b, hypotenuse c: $a^2 + b^2 = c^2$. Triples: 3-4-5, 5-12-13, 8-15-17	Chapter 9
Angle sums	Triangle $180°$; n-sided polygon $(n-2) \times 180°$; exterior angles of a convex polygon total $360°$	Chapter 9
Special right triangles	45-45-90: sides $1 : 1 : \sqrt{2}$. 30-60-90: sides $1 : \sqrt{3} : 2$	Chapter 9
Rectangle / parallelogram	Area lw; parallelogram bh (h perpendicular height)	Chapter 9
Triangle	Area $\frac{1}{2}bh = \frac{1}{2}ab\sin C$; Heron: $\sqrt{s(s-a)(s-b)(s-c)}$ with $s = \frac{a+b+c}{2}$	Chapter 9
Trapezoid (UK: trapezium)	Area $\frac{1}{2}(a+b)h$, a, b the parallel sides	Chapter 9
Circle	Circumference $2\pi r = \pi d$; area πr^2	Chapter 9
Arc and sector (radians)	Arc length $s = r\theta$; sector area $\frac{1}{2}r^2\theta$	Chapter 9
Degrees and radians	π rad $= 180°$; 1 rad $\approx 57.3°$	Chapter 13
Prism / cylinder	Volume Bh (B = base area); cylinder $\pi r^2 h$, surface area $2\pi r^2 + 2\pi r h$	Chapter 9
Pyramid / cone	Volume $\frac{1}{3}Bh$; cone $\frac{1}{3}\pi r^2 h$	Chapter 9
Sphere	Volume $\frac{4}{3}\pi r^3$; surface area $4\pi r^2$	Chapter 9
Similar figures	Scale lengths by k: areas scale by k^2, volumes by k^3	Chapter 9
Circle theorems	Angle in a semicircle is $90°$; inscribed angle is half the central angle on the same arc; tangent is perpendicular to the radius	Chapter 9

A.7. Trigonometry

Fact	Formula	Ch.
SOH-CAH-TOA	$\sin\theta = \frac{\text{opp}}{\text{hyp}}, \cos\theta = \frac{\text{adj}}{\text{hyp}}, \tan\theta = \frac{\text{opp}}{\text{adj}} = \frac{\sin\theta}{\cos\theta}$	Chapter 13
Reciprocal functions	$\sec\theta = \frac{1}{\cos\theta}, \csc\theta = \frac{1}{\sin\theta}, \cot\theta = \frac{1}{\tan\theta}$	Chapter 13
Unit circle	The point at angle θ is $(\cos\theta, \sin\theta)$	Chapter 13
Signs by quadrant ("All Students Take Calculus")	Q1 all positive; Q2 only $\sin$; Q3 only $\tan$; Q4 only $\cos$	Chapter 13
Pythagorean identities	$\sin^2\theta + \cos^2\theta = 1, 1 + \tan^2\theta = \sec^2\theta,$ $1 + \cot^2\theta = \csc^2\theta$	Chapter 13
Symmetry	$\sin(-\theta) = -\sin\theta, \cos(-\theta) = \cos\theta,$ $\tan(-\theta) = -\tan\theta$	Chapter 13
Co-function	$\sin\left(\frac{\pi}{2} - \theta\right) = \cos\theta, \cos\left(\frac{\pi}{2} - \theta\right) = \sin\theta$	Chapter 13
Periodicity	$\sin$ and $\cos$ have period 2π; $\tan$ has period π; $A\sin(bx + c) + d$ has amplitude $\lvert A\rvert$, period $2\pi/\lvert b\rvert$	Chapter 13
Angle addition	$\sin(\alpha \pm \beta) = \sin\alpha\cos\beta \pm \cos\alpha\sin\beta;$ $\cos(\alpha \pm \beta) = \cos\alpha\cos\beta \mp \sin\alpha\sin\beta$	Chapter 13
Tangent addition	$\tan(\alpha \pm \beta) = \frac{\tan\alpha \pm \tan\beta}{1 \mp \tan\alpha\tan\beta}$	Chapter 13
Double angle	$\sin 2\theta = 2\sin\theta\cos\theta;$ $\cos 2\theta = \cos^2\theta - \sin^2\theta = 2\cos^2\theta - 1 = 1 - 2\sin^2\theta$; $\tan 2\theta = \frac{2\tan\theta}{1 - \tan^2\theta}$	Chapter 13
Power reduction	$\sin^2\theta = \frac{1}{2}(1 - \cos 2\theta), \cos^2\theta = \frac{1}{2}(1 + \cos 2\theta)$	Chapter 13
Law of sines	$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}$ ($= 2R$, R the circumradius)	Chapter 13

Fact	Formula	Ch.
Law of cosines	$c^2 = a^2 + b^2 - 2ab\cos C$	Chapter 13
Inverse function ranges	$\arcsin : [-\tfrac{\pi}{2}, \tfrac{\pi}{2}]$, $\arccos : [0, \pi]$, $\arctan : (-\tfrac{\pi}{2}, \tfrac{\pi}{2})$	Chapter 13
Polar coordinates	$x = r\cos\theta, y = r\sin\theta; r = \sqrt{x^2 + y^2}$, $\theta = \operatorname{atan2}(y, x)$	Chapter 13
Rotation by θ	$(x, y) \mapsto (x\cos\theta - y\sin\theta,\ x\sin\theta + y\cos\theta)$	Chapter 13

Table 1. Exact values on the unit circle

Degrees	Radians	sin	cos	tan
$0°$	0	0	1	0
$30°$	$\tfrac{\pi}{6}$	$\tfrac{1}{2}$	$\tfrac{\sqrt{3}}{2}$	$\tfrac{1}{\sqrt{3}} = \tfrac{\sqrt{3}}{3}$
$45°$	$\tfrac{\pi}{4}$	$\tfrac{\sqrt{2}}{2}$	$\tfrac{\sqrt{2}}{2}$	1
$60°$	$\tfrac{\pi}{3}$	$\tfrac{\sqrt{3}}{2}$	$\tfrac{1}{2}$	$\sqrt{3}$
$90°$	$\tfrac{\pi}{2}$	1	0	undefined
$120°$	$\tfrac{2\pi}{3}$	$\tfrac{\sqrt{3}}{2}$	$-\tfrac{1}{2}$	$-\sqrt{3}$
$135°$	$\tfrac{3\pi}{4}$	$\tfrac{\sqrt{2}}{2}$	$-\tfrac{\sqrt{2}}{2}$	-1
$150°$	$\tfrac{5\pi}{6}$	$\tfrac{1}{2}$	$-\tfrac{\sqrt{3}}{2}$	$-\tfrac{\sqrt{3}}{3}$
$180°$	π	0	-1	0
$270°$	$\tfrac{3\pi}{2}$	-1	0	undefined
$360°$	2π	0	1	0

Memory aid for the first quadrant: $\sin$ at $0°, 30°, 45°, 60°, 90°$ is $\tfrac{\sqrt{0}}{2}, \tfrac{\sqrt{1}}{2}, \tfrac{\sqrt{2}}{2}, \tfrac{\sqrt{3}}{2}, \tfrac{\sqrt{4}}{2}$; $\cos$ is the same list backwards. See Chapter 13.

A.8. Sequences and series

Fact	Formula	Ch.
Arithmetic sequence	$a_n = a_1 + (n-1)d$; sum $S_n = \dfrac{n(a_1 + a_n)}{2} = \dfrac{n}{2}(2a_1 + (n-1)d)$	Chapter 14
Geometric sequence	$a_n = a_1 r^{n-1}$; sum $S_n = a_1 \dfrac{1 - r^n}{1 - r}$ $(r \neq 1)$	Chapter 14
Infinite geometric series	$\sum_{n=0}^{\infty} ar^n = \dfrac{a}{1-r}$ if $\lvert r \rvert < 1$; diverges otherwise (for $a \neq 0$)	Chapter 14
Sum of first n integers	$\sum_{k=1}^{n} k = \dfrac{n(n+1)}{2}$	Chapter 14
Sum of squares / cubes	$\sum_{k=1}^{n} k^2 = \dfrac{n(n+1)(2n+1)}{6}$, $\sum_{k=1}^{n} k^3 = \left(\dfrac{n(n+1)}{2}\right)^2$	Chapter 14
Powers of two	$\sum_{k=0}^{n-1} 2^k = 2^n - 1$	Chapter 14
Harmonic numbers	$H_n = \sum_{k=1}^{n} \dfrac{1}{k} \approx \ln n + 0.5772$; the harmonic series diverges	Chapter 14
p-series	$\sum_{n=1}^{\infty} \dfrac{1}{n^p}$ converges iff $p > 1$; $\sum \dfrac{1}{n^2} = \dfrac{\pi^2}{6}$	Chapter 14
Sigma rules	$\sum (a_k + b_k) = \sum a_k + \sum b_k$, $\sum c\,a_k = c \sum a_k$, $\sum_{k=1}^{n} c = nc$	Chapter 14
Recurrence (linear, first order)	$a_n = r a_{n-1} + d$ has fixed point $L = \dfrac{d}{1-r}$ and $a_n = L + r^n(a_0 - L)$ $(r \neq 1)$	Chapter 14
Taylor polynomial	$T_n(x) = \sum_{k=0}^{n} \dfrac{f^{(k)}(a)}{k!}(x-a)^k$	Volume 2, Chapter 18

Fact	Formula	Ch.
Standard series	$e^x = \sum_{n\geq 0} \frac{x^n}{n!}$; $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$; $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots$ (all x)	Volume 2, Chapter 18
More standard series	$\frac{1}{1-x} = \sum_{n\geq 0} x^n$ ($\lvert x\rvert < 1$); $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots$ ($-1 < x \leq 1$)	Volume 2, Chapter 18

A.9. Complex numbers

Fact	Formula	Ch.
Imaginary unit	$i^2 = -1$; powers cycle $i, -1, -i, 1$	Chapter 15
Arithmetic	$(a+bi) \pm (c+di) = (a \pm c) + (b \pm d)i$; $(a+bi)(c+di) = (ac-bd) + (ad+bc)i$	Chapter 15
Conjugate	$\overline{a+bi} = a-bi$; $z\bar{z} = a^2+b^2$; $\overline{zw} = \bar{z}\,\bar{w}$	Chapter 15
Division	$\dfrac{a+bi}{c+di} = \dfrac{(a+bi)(c-di)}{c^2+d^2} = \dfrac{(ac+bd)+(bc-ad)i}{c^2+d^2}$	Chapter 15
Modulus and argument	$\lvert z\rvert = \sqrt{a^2+b^2}$; $\arg z = \operatorname{atan2}(b, a)$	Chapter 15
Polar / exponential form	$z = r(\cos\theta + i\sin\theta) = re^{i\theta}$	Chapter 15
Euler's formula	$e^{i\theta} = \cos\theta + i\sin\theta$; hence $e^{i\pi} + 1 = 0$	Chapter 15
Multiplying in polar form	Moduli multiply, arguments add: $r_1 e^{i\theta_1} \cdot r_2 e^{i\theta_2} = r_1 r_2\, e^{i(\theta_1+\theta_2)}$	Chapter 15
De Moivre	$(\cos\theta + i\sin\theta)^n = \cos n\theta + i\sin n\theta$	Chapter 15
Roots of unity	Solutions of $z^n = 1$: $z_k = e^{2\pi i k/n}$, $k = 0, 1, \ldots, n-1$	Chapter 15
Real polynomials	Non-real roots come in conjugate pairs $a \pm bi$	Chapter 15

A.10. Limits and continuity

Fact	Formula	Ch.
Limit laws	Limits of sums, differences, products, constant multiples and quotients are the sums (etc.) of the limits, provided those exist and the denominator's limit is nonzero	Volume 2, Chapter 16
Continuity at a	$\lim_{x\to a} f(x) = f(a)$	Volume 2, Chapter 16
Formal definition	$\lim_{x\to a} f(x) = L$ means: for every $\varepsilon > 0$ there is $\delta > 0$ with $0 < \lvert x-a\rvert < \delta \implies \lvert f(x)-L\rvert < \varepsilon$	Volume 2, Chapter 16
Key trigonometric limits	$\lim_{x\to 0} \dfrac{\sin x}{x} = 1$, $\lim_{x\to 0} \dfrac{1-\cos x}{x} = 0$	Volume 2, Chapter 16
Limits defining e	$\lim_{n\to\infty} \left(1+\dfrac{1}{n}\right)^n = e$, $\lim_{x\to 0} \dfrac{e^x-1}{x} = 1$	Volume 2, Chapter 16
Growth hierarchy as $x \to \infty$	$\ln x \ll x^p \ll a^x \ll x!$ (any $p > 0$, $a > 1$); e.g. $\dfrac{x^{100}}{e^x} \to 0$	Volume 2, Chapter 16
Rational functions at ∞	Compare degrees: top lower $\to 0$; equal $\to$ ratio of leading coefficients; top higher $\to \pm\infty$	Volume 2, Chapter 16
Squeeze theorem	If $g \le f \le h$ near a and $g, h \to L$, then $f \to L$	Volume 2, Chapter 16

Fact	Formula	Ch.
L’Hôpital’s rule	For $\frac{0}{0}$ or $\frac{\infty}{\infty}$ forms: $\lim \frac{f(x)}{g(x)} = \lim \frac{f'(x)}{g'(x)}$ if the right side exists	Volume 2, Chapter 18
Intermediate value theorem	If f is continuous on $[a, b]$ and y lies between $f(a)$ and $f(b)$, some c in $[a, b]$ has $f(c) = y$	Volume 2, Chapter 16

A.11. Derivatives

Rule	Formula	Ch.
Definition	$f'(x) = \lim_{h \to 0} \dfrac{f(x+h) - f(x)}{h}$	Volume 2, Chapter 17
Constant / linearity	$(c)' = 0$; $(af + bg)' = af' + bg'$	Volume 2, Chapter 17
Power rule	$\dfrac{d}{dx} x^n = n x^{n-1}$ (any real n)	Volume 2, Chapter 17
Product rule	$(fg)' = f'g + fg'$	Volume 2, Chapter 17
Quotient rule	$\left(\dfrac{f}{g}\right)' = \dfrac{f'g - fg'}{g^2}$	Volume 2, Chapter 17
Chain rule	$\dfrac{d}{dx} f(g(x)) = f'(g(x))\, g'(x)$; in Leibniz form $\dfrac{dy}{dx} = \dfrac{dy}{du}\dfrac{du}{dx}$	Volume 2, Chapter 17
Exponentials	$(e^x)' = e^x$; $(a^x)' = a^x \ln a$	Volume 2, Chapter 17
Logarithms	$(\ln x)' = \dfrac{1}{x}$; $(\log_a x)' = \dfrac{1}{x \ln a}$; $(\ln \lvert x \rvert)' = \dfrac{1}{x}$	Volume 2, Chapter 17

Rule	Formula	Ch.
Trigonometric	$(\sin x)' = \cos x$; $(\cos x)' = -\sin x$; $(\tan x)' = \sec^2 x$	Volume 2, Chapter 17
Inverse trigonometric	$(\arcsin x)' = \frac{1}{\sqrt{1-x^2}}$; $(\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$; $(\arctan x)' = \frac{1}{1+x^2}$	Volume 2, Chapter 17
Logistic sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$, $\sigma'(x) = \sigma(x)\,(1-\sigma(x))$	Volume 3, Chapter 33
Implicit differentiation	Differentiate both sides with respect to x, treating y as $y(x)$: e.g. $x^2 + y^2 = 1 \implies 2x + 2yy' = 0 \implies y' = -x/y$	Volume 2, Chapter 17
Tangent line / linearisation	$y = f(a) + f'(a)(x-a)$; so $f(a+h) \approx f(a) + hf'(a)$	Volume 2, Chapter 18
Critical points and tests	Critical point: $f'(c) = 0$ or undefined. Second-derivative test: $f''(c) > 0$ local min, $f''(c) < 0$ local max, $= 0$ inconclusive	Volume 2, Chapter 18
Closed-interval extrema	On $[a, b]$ check critical points and both endpoints; the largest/smallest value wins	Volume 2, Chapter 18
Concavity / inflection	$f'' > 0$ concave up, $f'' < 0$ concave down; inflection where f'' changes sign	Volume 2, Chapter 18

Rule	Formula	Ch.
Mean value theorem	f continuous on $[a, b]$, differentiable inside $\implies f'(c) = \dfrac{f(b) - f(a)}{b - a}$ for some $c \in (a, b)$	Volume 2, Chapter 18
Newton's method	$x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$	Volume 2, Chapter 18

A.12. Integrals

Rule	Formula	Ch.
Power rule	$\int x^n\,dx = \frac{x^{n+1}}{n+1} + C\,(n \neq -1)$	Volume 2, Chapter 19
Reciprocal	$\int \frac{1}{x}\,dx = \ln\lvert x\rvert + C$	Volume 2, Chapter 19
Exponentials	$\int e^x\,dx = e^x + C;\int e^{kx}\,dx = \frac{1}{k}e^{kx} + C;$ $\int a^x\,dx = \frac{a^x}{\ln a} + C$	Volume 2, Chapter 19
Trigonometric	$\int \sin x\,dx = -\cos x + C;\int \cos x\,dx = \sin x + C;$ $\int \sec^2 x\,dx = \tan x + C$	Volume 2, Chapter 19
Inverse-trig forms	$\int \frac{dx}{1+x^2} = \arctan x + C;$ $\int \frac{dx}{\sqrt{1-x^2}} = \arcsin x + C$	Volume 2, Chapter 19
Logarithm	$\int \ln x\,dx = x\ln x - x + C$	Volume 2, Chapter 19
Linearity	$\int (af + bg)\,dx = a\int f\,dx + b\int g\,dx$	Volume 2, Chapter 19
Fundamental theorem (part 1)	$\frac{d}{dx}\int_a^x f(t)\,dt = f(x)$	Volume 2, Chapter 19

Rule	Formula	Ch.
Fundamental theorem (part 2)	$\int_a^b f(x)\,dx = F(b) - F(a)$ where $F' = f$	Volume 2, Chapter 19
Definite-integral properties	$\int_a^a f = 0$; $\int_b^a f = -\int_a^b f$; $\int_a^b f + \int_b^c f = \int_a^c f$	Volume 2, Chapter 19
Substitution	$\int f(g(x))\,g'(x)\,dx = \int f(u)\,du$ with $u = g(x)$; for definite integrals change the limits to $g(a), g(b)$	Volume 2, Chapter 19
Integration by parts	$\int u\,dv = uv - \int v\,du$	Volume 2, Chapter 19
Riemann sum	$\int_a^b f(x)\,dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i^*)\,\Delta x$, $\Delta x = \dfrac{b-a}{n}$	Volume 2, Chapter 19
Area between curves	$\int_a^b (f(x) - g(x))dx$ where $f \geq g$ on $[a, b]$	Volume 2, Chapter 19
Average value	$\bar{f} = \dfrac{1}{b-a}\int_a^b f(x)\,dx$	Volume 2, Chapter 19
Volume of revolution (discs)	$V = \pi \int_a^b f(x)^2\,dx$ about the x-axis	Volume 2, Chapter 19

Rule	Formula	Ch.
Gaussian integral	$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}$	Volume 3, Chapter 30

A.13. Multivariable calculus

Fact	Formula	Ch.
Partial derivative	$\dfrac{\partial f}{\partial x}$: differentiate in x holding the other variables fixed. Example: $f = x^2 y \implies f_x = 2xy,\ f_y = x^2$	Volume 2, Chapter 20
Gradient	$\nabla f = \left(\dfrac{\partial f}{\partial x_1}, \dots, \dfrac{\partial f}{\partial x_n} \right)$; points in the direction of steepest ascent; its length is that steepest rate	Volume 2, Chapter 20
Directional derivative	$D_{\mathbf{u}} f = \nabla f \cdot \mathbf{u}$ for a unit vector $\mathbf{u}$	Volume 2, Chapter 20
Gradient descent	$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla f(\mathbf{x}_k)$ (η the learning rate / step size)	Volume 2, Chapter 20
Tangent plane	$z = f(a,b) + f_x(a,b)(x-a) + f_y(a,b)(y-b)$	Volume 2, Chapter 20
Multivariable chain rule	$\dfrac{d}{dt} f(x(t), y(t)) = f_x \dfrac{dx}{dt} + f_y \dfrac{dy}{dt} = \nabla f \cdot \mathbf{r}'(t)$	Volume 2, Chapter 20
Critical points and second-derivative test	$\nabla f = \mathbf{0}$; with $D = f_{xx} f_{yy} - f_{xy}^2$: $D > 0, f_{xx} > 0$ min; $D > 0, f_{xx} < 0$ max; $D < 0$ saddle; $D = 0$ inconclusive	Volume 2, Chapter 20
Hessian	$H_{ij} = \dfrac{\partial^2 f}{\partial x_i \partial x_j}$; symmetric for smooth f	Volume 2, Chapter 20

Fact	Formula	Ch.
Lagrange multipliers	Optimise f subject to $g = c$: solve $\nabla f = \lambda \nabla g$, $g = c$	Volume 2, Chapter 20

A.14. Logic

Law	Equivalence	Ch.
Conditional as "or"	$p \to q \equiv \neg p \lor q$; false only when p is true and q false	Volume 3, Chapter 21
Contrapositive (equivalent)	$p \to q \equiv \neg q \to \neg p$	Volume 3, Chapter 21
Converse and inverse (not equivalent to $p \to q$)	Converse $q \to p$; inverse $\neg p \to \neg q$ (these two are equivalent to each other)	Volume 3, Chapter 21
Negating a conditional	$\neg(p \to q) \equiv p \land \neg q$	Volume 3, Chapter 21
Biconditional	$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$	Volume 3, Chapter 21
De Morgan	$\neg(p \land q) \equiv \neg p \lor \neg q$; $\neg(p \lor q) \equiv \neg p \land \neg q$	Volume 3, Chapter 21
Double negation	$\neg\neg p \equiv p$	Volume 3, Chapter 21
Distributive	$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$; $p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$	Volume 3, Chapter 21

Law	Equivalence	Ch.
Absorption	$p \lor (p \land q) \equiv p$; $p \land (p \lor q) \equiv p$	Volume 3, Chapter 21
Excluded middle / contradiction	$p \lor \neg p \equiv \mathbf{T}$; $p \land \neg p \equiv \mathbf{F}$	Volume 3, Chapter 21
Exclusive or	$p \oplus q \equiv (p \lor q) \land \neg(p \land q)$	Volume 3, Chapter 21
Negating quantifiers	$\neg \forall x\, P(x) \equiv \exists x\, \neg P(x)$; $\neg \exists x\, P(x) \equiv \forall x\, \neg P(x)$	Volume 3, Chapter 21
Quantifier order matters	$\forall x\, \exists y\, P(x, y)$ is generally *not* equivalent to $\exists y\, \forall x\, P(x, y)$	Volume 3, Chapter 21

A.15. Sets

Identity	Formula	Ch.
Basic operations	$A \cup B$ (in either), $A \cap B$ (in both), $A \setminus B$ (in A not B), A^c (not in A)	Volume 3, Chapter 21
Difference via complement	$A \setminus B = A \cap B^c$	Volume 3, Chapter 21
De Morgan for sets	$(A \cup B)^c = A^c \cap B^c$; $(A \cap B)^c = A^c \cup B^c$	Volume 3, Chapter 21
Distributive	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$; $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Volume 3, Chapter 21
Subset tests	$A \subseteq B \iff A \cap B = A \iff A \cup B = B$	Volume 3, Chapter 21
Inclusion-exclusion (two sets)	$\lvert A \cup B \rvert = \lvert A \rvert + \lvert B \rvert - \lvert A \cap B \rvert$	Volume 3, Chapter 24
Inclusion-exclusion (three sets)	$\lvert A \cup B \cup C \rvert = \lvert A \rvert + \lvert B \rvert + \lvert C \rvert - \lvert A \cap B \rvert - \lvert A \cap C \rvert - \lvert B \cap C \rvert + \lvert A \cap B \cap C \rvert$	Volume 3, Chapter 24
Power set	$\lvert \mathcal{P}(A) \rvert = 2^{\lvert A \rvert}$	Volume 3, Chapter 21

Identity	Formula	Ch.
Cartesian product	$\lvert A \times B \rvert = \lvert A \rvert \cdot \lvert B \rvert$	Volume 3, Chapter 21

A.16. Proof techniques

Technique	To prove … do this	Ch.
Direct proof	$p \to q$: assume p, deduce q step by step	Volume 3, Chapter 22
Contrapositive	$p \to q$: assume $\neg q$, deduce $\neg p$	Volume 3, Chapter 22
Contradiction	p: assume $\neg p$, derive something impossible	Volume 3, Chapter 22
Cases	Split into exhaustive cases and prove each	Volume 3, Chapter 22
Counterexample	To *disprove* $\forall x\, P(x)$, exhibit one x with $\neg P(x)$	Volume 3, Chapter 22
Induction	$P(n)$ for all $n \geq n_0$: prove $P(n_0)$ (base case), then $P(k) \implies P(k+1)$ (inductive step)	Volume 3, Chapter 22
Strong induction	Inductive step assumes $P(n_0), \ldots, P(k)$ all hold, then proves $P(k+1)$	Volume 3, Chapter 22
Loop invariant	Induction applied to code: true before the loop, preserved by each iteration, so true at exit	Volume 3, Chapter 22

A.17. Counting

Situation	Count	Ch.
Sum rule	Choose one of m options *or* one of n (disjoint): $m + n$	Volume 3, Chapter 24
Product rule	Make a choice with m options *then* one with n: mn	Volume 3, Chapter 24
Ordered, repetition allowed	Sequences of length r from n symbols: n^r	Volume 3, Chapter 24
Arrangements of n distinct items	$n! = n(n-1)\cdots 2 \cdot 1$, with $0! = 1$	Volume 3, Chapter 24
Permutations (ordered, no repetition)	$P(n, r) = \dfrac{n!}{(n-r)!}$	Volume 3, Chapter 24
Combinations (unordered, no repetition)	$\dbinom{n}{r} = \dfrac{n!}{r!\,(n-r)!}$	Volume 3, Chapter 24
Multisets / stars and bars (unordered, repetition allowed)	Choose r from n types: $\dbinom{n+r-1}{r}$	Volume 3, Chapter 24
Arrangements with repeated items	$\dfrac{n!}{n_1!\,n_2!\cdots n_k!}$ (e.g. MISSISSIPPI: $\frac{11!}{1!\,4!\,4!\,2!} = 34650$)	Volume 3, Chapter 24

Situation	Count	Ch.
Pascal’s rule and symmetry	$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$; $\binom{n}{k} = \binom{n}{n-k}$	Volume 3, Chapter 24
Row sum	$\sum_{k=0}^{n} \binom{n}{k} = 2^n$ (the number of subsets of an n-set)	Volume 3, Chapter 24
Derangements	Permutations with no fixed point: $D_n = n!\sum_{k=0}^{n} \frac{(-1)^k}{k!} \approx \frac{n!}{e}$	Volume 3, Chapter 24
Pigeonhole principle	n items in $m < n$ boxes: some box holds at least $\lceil n/m \rceil$ items	Volume 3, Chapter 24

A.18. Graphs

Fact	Formula	Ch.
Handshake lemma	$\sum_v \deg(v) = 2\lvert E\rvert$; so the number of odd-degree vertices is even	Volume 3, Chapter 25
Complete graph	K_n has $\binom{n}{2} = \frac{n(n-1)}{2}$ edges	Volume 3, Chapter 25
Trees	A tree on n vertices is connected, acyclic, and has exactly $n - 1$ edges	Volume 3, Chapter 25
Euler circuit	Exists in a connected graph iff every vertex has even degree (path: exactly 0 or 2 odd vertices)	Volume 3, Chapter 25
Euler's formula (planar, connected)	$V - E + F = 2$	Volume 3, Chapter 25
Traversal cost	BFS and DFS run in $O(V + E)$ with adjacency lists	Volume 3, Chapter 25

A.19. Modular arithmetic and number theory

Fact	Formula	Ch.
Congruence	$a \equiv b \pmod{m} \iff m \mid (a-b)$ (same remainder on division by m)	Volume 3, Chapter 23
Arithmetic respects congruence	If $a \equiv b$ and $c \equiv d \pmod{m}$ then $a \pm c \equiv b \pm d$ and $ac \equiv bd \pmod{m}$	Volume 3, Chapter 23
Reduce early	$(a \cdot b) \bmod m = ((a \bmod m)(b \bmod m)) \bmod m$; same for $+$	Volume 3, Chapter 23
Euclidean algorithm	$\gcd(a, b) = \gcd(b, a \bmod b)$, $\gcd(a, 0) = a$	Volume 3, Chapter 23
Bézout's identity	$\gcd(a, b) = ax + by$ for some integers x, y (found by the extended Euclidean algorithm)	Volume 3, Chapter 23
Modular inverse	$a^{-1} \bmod m$ exists iff $\gcd(a, m) = 1$	Volume 3, Chapter 23
Fundamental theorem of arithmetic	Every integer $n \geq 2$ factors into primes uniquely (up to order)	Volume 3, Chapter 23
Fermat's little theorem	p prime, $p \nmid a \implies a^{p-1} \equiv 1 \pmod{p}$	Volume 3, Chapter 23

Fact	Formula	Ch.
Euler’s totient	$\varphi(n)$ = number of $1 \le k \le n$ coprime to n; $\varphi(p) = p-1$; $\varphi(pq) = (p-1)(q-1)$ for distinct primes	Volume 3, Chapter 23
Euler’s theorem	$\gcd(a,n)=1 \implies a^{\varphi(n)} \equiv 1 \pmod{n}$	Volume 3, Chapter 23
Chinese remainder theorem	If $m_1, \dots, m_k$ are pairwise coprime, the system $x \equiv a_i \pmod{m_i}$ has a unique solution modulo $m_1 m_2 \cdots m_k$	Volume 3, Chapter 23
RSA	$n = pq$; pick e coprime to $\varphi(n)$; $d \equiv e^{-1} \pmod{\varphi(n)}$. Encrypt $c = m^e \bmod n$; decrypt $m = c^d \bmod n$	Volume 3, Chapter 23
Fast exponentiation	Square-and-multiply computes $a^k \bmod m$ in $O(\log k)$ multiplications	Volume 3, Chapter 23

A.20. Vectors and matrices

Fact	Formula	Ch.
Length (norm)	$\|\mathbf{v}\| = \sqrt{v_1^2 + \cdots + v_n^2}$; unit vector $\hat{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$	Volume 2, Chapter 26
Dot product	$\mathbf{u}\cdot\mathbf{v} = \sum_i u_i v_i = \|\mathbf{u}\|\|\mathbf{v}\|\cos\theta$	Volume 2, Chapter 26
Orthogonality	$\mathbf{u}\perp\mathbf{v} \iff \mathbf{u}\cdot\mathbf{v} = 0$	Volume 2, Chapter 26
Cosine similarity	$\cos\theta = \dfrac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$	Volume 2, Chapter 26
Projection of $\mathbf{u}$ onto $\mathbf{v}$	$\operatorname{proj}_{\mathbf{v}}\mathbf{u} = \dfrac{\mathbf{u}\cdot\mathbf{v}}{\mathbf{v}\cdot\mathbf{v}}\,\mathbf{v}$	Volume 2, Chapter 26
Cross product (3D)	$\mathbf{u}\times\mathbf{v} = (u_2v_3 - u_3v_2,\ u_3v_1 - u_1v_3,\ u_1v_2 - u_2v_1)$; length $\|\mathbf{u}\|\|\mathbf{v}\|\sin\theta$, perpendicular to both	Volume 2, Chapter 26
Triangle and Cauchy-Schwarz inequalities	$\|\mathbf{u}+\mathbf{v}\| \le \|\mathbf{u}\| + \|\mathbf{v}\|$; $\|\mathbf{u}\cdot\mathbf{v}\| \le \|\mathbf{u}\|\|\mathbf{v}\|$	Volume 2, Chapter 26
Matrix product	$(AB)_{ij} = \sum_k A_{ik}B_{kj}$; needs (columns of A) = (rows of B); in general $AB \neq BA$	Volume 2, Chapter 27

Fact	Formula	Ch.
Transpose rules	$(A^T)_{ij} = A_{ji}$; $(AB)^T = B^T A^T$	Volume 2, Chapter 27
Inverse rules	$AA^{-1} = A^{-1}A = I$; $(AB)^{-1} = B^{-1}A^{-1}$; $(A^T)^{-1} = (A^{-1})^T$	Volume 2, Chapter 27
2×2 determinant	$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$	Volume 2, Chapter 27
2×2 inverse	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$ (if $ad - bc \neq 0$)	Volume 2, Chapter 27
3×3 determinant (first-row expansion)	$a(ei - fh) - b(di - fg) + c(dh - eg)$ for rows $(a,b,c), (d,e,f), (g,h,i)$	Volume 2, Chapter 27
Determinant facts	$\det(AB) = \det A \det B$; $\det A^T = \det A$; A invertible $\iff \det A \neq 0$; $\lvert\det A\rvert$ is the area/volume scale factor	Volume 2, Chapter 27
Rotation by θ (2D)	$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$; $R_\theta^{-1} = R_\theta^T = R_{-\theta}$	Volume 2, Chapter 27
Least squares (normal equations)	Minimise $\lVert A\mathbf{x} - \mathbf{b}\rVert$: solve $A^T A\mathbf{x} = A^T\mathbf{b}$	Volume 2, Chapter 27

Fact	Formula	Ch.
Eigenvalues and eigenvectors	$A\mathbf{v} = \lambda\mathbf{v}$, $\mathbf{v} \neq \mathbf{0}$; eigenvalues solve $\det(A - \lambda I) = 0$	Volume 2, Chapter 28
2×2 characteristic equation	$\lambda^2 - (\operatorname{tr} A)\lambda + \det A = 0$; in general $\operatorname{tr} A = \sum \lambda_i$, $\det A = \prod \lambda_i$	Volume 2, Chapter 28
Diagonalisation / spectral theorem	$A = PDP^{-1}$; for symmetric A, $A = Q\Lambda Q^T$ with Q orthogonal and real eigenvalues	Volume 2, Chapter 28
Singular value decomposition	$A = U\Sigma V^T$ (U, V orthogonal, Σ diagonal, non-negative)	Volume 2, Chapter 28

A.21. Probability

Rule	Formula	Ch.
Range and total	$0 \le P(A) \le 1$, $P(\Omega) = 1$	Volume 3, Chapter 29
Equally likely outcomes	$P(A) = \dfrac{\text{number of outcomes in } A}{\text{total number of outcomes}}$	Volume 3, Chapter 29
Complement	$P(A^c) = 1 - P(A)$; "at least one" $= 1 - P(\text{none})$	Volume 3, Chapter 29
Addition rule	$P(A \cup B) = P(A) + P(B) - P(A \cap B)$; just $P(A) + P(B)$ if mutually exclusive	Volume 3, Chapter 29
Conditional probability	$P(A \mid B) = \dfrac{P(A \cap B)}{P(B)}$ $(P(B) > 0)$	Volume 3, Chapter 29
Multiplication rule	$P(A \cap B) = P(A \mid B)\,P(B) = P(B \mid A)\,P(A)$	Volume 3, Chapter 29
Independence	$P(A \cap B) = P(A)\,P(B)$, equivalently $P(A \mid B) = P(A)$	Volume 3, Chapter 29
Law of total probability	$P(A) = \sum_i P(A \mid B_i)\,P(B_i)$ for a partition $B_1, B_2, \ldots$	Volume 3, Chapter 29

Rule	Formula	Ch.
Bayes' theorem	$P(H \mid E) = \frac{P(E \mid H)\,P(H)}{P(E)} = \frac{P(E \mid H)\,P(H)}{P(E \mid H)\,P(H) + P(E \mid H^c)\,P(H^c)}$	Volume 3, Chapter 29
Birthday bound	With N equally likely values and n draws, $P(\text{some repeat}) \approx 1 - e^{-n(n-1)/(2N)}$; likely once $n \approx 1.18\sqrt{N}$	Volume 3, Chapter 29
Expectation	$E[X] = \sum_x x\,P(X = x)$ (discrete) or $\int x\,f(x)\,dx$ (continuous)	Volume 3, Chapter 30
Linearity of expectation	$E[aX + bY + c] = aE[X] + bE[Y] + c$ (always, even if dependent)	Volume 3, Chapter 30
Variance	$\mathrm{Var}(X) = E[(X - \mu)^2] = E[X^2] - (E[X])^2$; $\sigma = \sqrt{\mathrm{Var}(X)}$	Volume 3, Chapter 30
Variance rules	$\mathrm{Var}(aX + b) = a^2\,\mathrm{Var}(X)$; $\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\,\mathrm{Cov}(X, Y)$	Volume 3, Chapter 30
Covariance and correlation	$\mathrm{Cov}(X, Y) = E[XY] - E[X]E[Y]$; $\rho = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y} \in [-1, 1]$; independent $\Longrightarrow$ covariance 0 (not conversely)	Volume 3, Chapter 30

A.22. Common distributions

Distribution	PMF or PDF	Mean	Variance	Ch.
Bernoulli (p)	$P(X=1)=p$, $P(X=0)=1-p$	p	$p(1-p)$	Volume 3, Chapter 30
Binomial (n,p)	$\binom{n}{k}p^k(1-p)^{n-k}$, $k=0,\dots,n$	np	$np(1-p)$	Volume 3, Chapter 30
Geometric (p) (trials until first success)	$(1-p)^{k-1}p$, $k=1,2,\dots$	$\frac{1}{p}$	$\frac{1-p}{p^2}$	Volume 3, Chapter 30
Poisson (λ)	$\frac{e^{-\lambda}\lambda^k}{k!}$, $k=0,1,2,\dots$	λ	λ	Volume 3, Chapter 30
Discrete uniform on $\{1,\dots,n\}$	$\frac{1}{n}$	$\frac{n+1}{2}$	$\frac{n^2-1}{12}$	Volume 3, Chapter 30
Continuous uniform on $[a,b]$	$\frac{1}{b-a}$ for $a\le x\le b$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$	Volume 3, Chapter 30
Exponential (λ)	$\lambda e^{-\lambda x}$ for $x\ge 0$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	Volume 3, Chapter 30
Normal (μ,σ^2)	$\frac{1}{\sigma\sqrt{2\pi}}\,e^{-(x-\mu)^2/(2\sigma^2)}$	μ	σ^2	Volume 3, Chapter 30

Fact	Formula	Ch.
Standardising	$Z = \frac{X - \mu}{\sigma}$ has mean 0 and standard deviation 1	Volume 3, Chapter 30
68-95-99.7 rule	For a normal distribution, about 68%, 95% and 99.7% of values lie within 1, 2 and 3 standard deviations of the mean	Volume 3, Chapter 30
Central limit theorem	For n independent draws with mean μ and variance σ^2, the sample mean $\bar{X}$ is approximately $N(\mu, \sigma^2/n)$ for large n	Volume 3, Chapter 30
Law of large numbers	$\bar{X} \to \mu$ as $n \to \infty$	Volume 3, Chapter 30

A.23. Statistics

Quantity	Formula	Ch.
Mean	$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$	Volume 3, Chapter 31
Median, quartiles, IQR	Median: middle value of the sorted data (average the two middle values if n is even); $\text{IQR} = Q_3 - Q_1$; outlier rule of thumb: beyond $1.5 \times \text{IQR}$ from the quartiles	Volume 3, Chapter 31
Population variance	$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$	Volume 3, Chapter 31
Sample variance	$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$ (divide by $n-1$, Bessel's correction)	Volume 3, Chapter 31
z-score	$z = \frac{x - \bar{x}}{s}$ (or $\frac{x-\mu}{\sigma}$)	Volume 3, Chapter 31
Standard error of the mean	$\text{SE} = \frac{\sigma}{\sqrt{n}} \approx \frac{s}{\sqrt{n}}$	Volume 3, Chapter 31
Confidence interval for a mean	$\bar{x} \pm t^* \frac{s}{\sqrt{n}}$ ($t^* \approx z^* = 1.96$ for 95% with large n)	Volume 3, Chapter 31
Confidence interval for a proportion	$\hat{p} \pm z^* \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$	Volume 3, Chapter 31

Quantity	Formula	Ch.
Sample size for margin of error E	Proportion: $n \approx \left(\frac{z^*}{E}\right)^2 \hat{p}(1-\hat{p})$ (use $\hat{p} = 0.5$ if unknown); 95%, $\pm 3\%$ needs about 1068	Volume 3, Chapter 31
Correlation coefficient	$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \in [-1, 1]$	Volume 3, Chapter 31
Least-squares regression line	$\hat{y} = a + bx$ with $b = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} = r\frac{s_y}{s_x}$, $a = \bar{y} - b\bar{x}$	Volume 3, Chapter 31
Coefficient of determination	$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$; equals r^2 for a straight-line fit	Volume 3, Chapter 31
Hypothesis test	p-value = probability, if the null hypothesis were true, of data at least as extreme as observed. Reject H_0 if p-value $< \alpha$ (often 0.05)	Volume 3, Chapter 31
Test statistic for a mean	$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$ ($n - 1$ degrees of freedom)	Volume 3, Chapter 31

A.24. Numerical methods

Method	Formula	Ch.
Newton's method	$x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$; quadratic convergence near a simple root (correct digits roughly double each step); can diverge from a poor start or where $f' \approx 0$	Volume 2, Chapter 32
Square root by Newton (Heron)	$x_{n+1} = \frac{1}{2}\left(x_n + \dfrac{a}{x_n}\right) \to \sqrt{a}$	Volume 2, Chapter 18
Bisection	If $f(a)$ and $f(b)$ differ in sign, test the midpoint $m = \frac{a+b}{2}$ and keep the half where the sign changes. After n halvings the bracket has width $(b-a)/2^n$; need about $\log_2\left((b-a)/\text{tol}\right)$ steps	Volume 2, Chapter 32
Secant method	$x_{n+1} = x_n - f(x_n)\dfrac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$ (Newton without a derivative)	Volume 2, Chapter 32
Fixed-point iteration	$x_{n+1} = g(x_n)$ converges near a fixed point x^* if $\lvert g'(x^*)\rvert < 1$	Volume 2, Chapter 32
Finite differences	Forward $f'(x) \approx \dfrac{f(x+h) - f(x)}{h}$ (error $O(h)$); central $f'(x) \approx \dfrac{f(x+h) - f(x-h)}{2h}$ (error $O(h^2)$)	Volume 2, Chapter 32
Trapezoid rule	$\int_a^b f \approx \dfrac{h}{2}(f_0 + 2f_1 + 2f_2 + \cdots + 2f_{n-1} + f_n)$, $h = \dfrac{b-a}{n}$, $f_i = f(a+ih)$; error $O(h^2)$	Volume 2, Chapter 32
Simpson's rule (n even)	$\int_a^b f \approx \dfrac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{n-2} + 4f_{n-1} + f_n)$; error $O(h^4)$	Volume 2, Chapter 32

Method	Formula	Ch.
Euler's method (ODE $y' = f(t, y)$)	$y_{n+1} = y_n + h\, f(t_n, y_n)$; error $O(h)$	Volume 2, Chapter 32
Floating point (IEEE double)	Machine epsilon $2^{-52} \approx 2.2 \times 10^{-16}$; about 15-16 significant decimal digits; never test floats with $==$, use a tolerance	Volume 2, Chapter 32
Absolute and relative error	$\lvert\hat{x} - x\rvert$ and $\dfrac{\lvert\hat{x} - x\rvert}{\lvert x\rvert}$	Volume 2, Chapter 32

A.25. Optimisation, machine learning, and information

Quantity	Formula	Ch.
Mean squared error	$\mathrm{MSE} = \dfrac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	Volume 3, Chapter 33
Softmax	$\mathrm{softmax}(\mathbf{z})_i = \dfrac{e^{z_i}}{\sum_j e^{z_j}}$ (subtract $\max_j z_j$ first for numerical safety)	Volume 3, Chapter 33
Entropy (bits)	$H(p) = -\sum_i p_i \log_2 p_i$; maximal ($\log_2 n$) for the uniform distribution on n outcomes	Volume 3, Chapter 33
Cross-entropy	$H(p, q) = -\sum_i p_i \log q_i$; with a one-hot label it is $-\log q_{\text{true}}$	Volume 3, Chapter 33
Kullback-Leibler divergence	$D_{\mathrm{KL}}(p \,\|\, q) = \sum_i p_i \log \dfrac{p_i}{q_i} = H(p, q) - H(p) \geq 0$	Volume 3, Chapter 33

Appendix B: Glossary

Every Refresher box in Volume 1, alphabetised. This is the book's reference mode: the same definitions you met in context, collected for lookup. Each entry links back to the chapter where it is taught.

B.1.

2D rotation formula (Chapter 13)

Rotating (x, y) anticlockwise about the origin by θ:

$$x' = x\cos\theta - y\sin\theta, \qquad y' = x\sin\theta + y\cos\theta.$$

In Volume 2, Chapter 27 this becomes multiplication by the **rotation matrix** $\begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}$. To rotate about some other point (c_x, c_y), translate it to the origin, rotate, and translate back.

B.2. A

Absolute value (Chapter 1)

The **absolute value** (or **magnitude**, or **modulus**) of x, written $|x|$, is its distance from 0:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x < 0. \end{cases}$$

The large brace introduces a **piecewise definition**: read each line as "this value, under this condition". So $|5| = 5$, $|-5| = -(-5) = 5$, $|0| = 0$. The distance between a and b on the line is $|a - b|$.

Absolute value as distance (Chapter 5)

The **absolute value** $|x|$ is the distance from x to 0 on the number line, so it is never negative: $|5| = 5$, $|-5| = 5$, $|0| = 0$. Formally,

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x < 0. \end{cases}$$

(The second line is not a typo: if x is negative, $-x$ is positive.) More usefully, $|x - a|$ is the **distance between x and a**. Read $|x - 3| < 2$ as "x is less than 2 away from 3".

Absolute value function (Chapter 11)

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

It is the distance from x to 0 on the number line, so it is never negative. $|x - a|$ is the distance between x and a. The graph is a V with its corner at the origin.

Analytic geometry (Chapter 10)

Analytic geometry (also **coordinate geometry** or **Cartesian geometry**) is the study of geometry using a coordinate system, so that geometric facts are proved and computed with algebra. It contrasts with **synthetic geometry** — Euclid's style in Chapter 9 — which reasons directly from axioms and diagrams without numbers.

Angle and its measure (Chapter 9)

An **angle** is formed by two rays with a common endpoint, the **vertex**. The angle with vertex B and rays through A and C is written $\angle ABC$ (vertex in the middle) or just $\angle B$. Its **measure** is written $m\angle ABC$ in US textbooks, though most people just write $\angle ABC = 40°$.

Angles are measured in **degrees**: a full turn is $360°$. (In Chapter 13 you'll meet **radians**, where a full turn is 2π.)

Angle pairs (Chapter 9)

- **Complementary** angles add to $90°$. **Supplementary** angles add to $180°$.
- **Adjacent** angles share a vertex and a side but don't overlap.
- A **linear pair** is two adjacent angles whose outer sides form a straight line; they are supplementary.
- **Vertical angles** (UK: **vertically opposite angles**) are the non-adjacent angles formed when two lines cross.

Angle-sum and difference formulas (Chapter 13)

$$\begin{aligned}\sin(\alpha \pm \beta) &= \sin\alpha\cos\beta \pm \cos\alpha\sin\beta \\ \cos(\alpha \pm \beta) &= \cos\alpha\cos\beta \mp \sin\alpha\sin\beta \\ \tan(\alpha \pm \beta) &= \frac{\tan\alpha \pm \tan\beta}{1 \mp \tan\alpha\tan\beta}\end{aligned}$$

$\pm$ means "plus or minus" and $\mp$ means "the opposite sign": take the top signs together or the bottom signs together. The cosine formula **flips** the sign.

Arithmetic sequence (Chapter 14)

An **arithmetic sequence** (or **arithmetic progression**, AP) adds a fixed **common difference** d each step: $a_n = a_{n-1} + d$. Its explicit form, with first term a_1, is $a_n = a_1 + (n-1)d$. The word is pronounced a-rith-MET-ic (stress on the third syllable) when used as an adjective.

Associativity of operators (left/right) (Chapter 1)

When operators of equal rank appear in a row, **operator associativity** decides grouping. Subtraction and division are **left-associative**: $a - b - c = (a - b) - c$. Exponentiation is **right-associative**: $a^{b^c} = a^{(b^c)}$, so $2^{3^2} = 2^9 = 512$, not $8^2 = 64$. (This is a different meaning of "associative" from the law in Section 1.4: here it is a parsing rule for notation.)

Asymptote (Chapter 12)

An **asymptote** is a line that a curve approaches more and more closely without reaching it (at least far out). $y = 0$ is a **horizontal asymptote** of $y = 2^x$: as x becomes very negative, 2^x gets as close to 0 as you like but stays positive.

Axiom, postulate, theorem, lemma, corollary (Chapter 9)

An **axiom** or **postulate** is a statement accepted without proof (Euclid used "postulate" for geometric ones, "common notion" for general logic such as "things equal to the same thing are equal to each other").

A **theorem** is a statement proved from axioms and earlier theorems. A **lemma** is a small theorem proved mainly as a step towards a bigger one. A **corollary** is a result that follows almost immediately from a theorem.

A **conjecture** is a statement believed true but not yet proved.

Axis of symmetry (Chapter 8)

The **axis of symmetry** is the vertical line through the vertex, $x = -\dfrac{b}{2a}$. The parabola is its own mirror image in this line. Consequence: the two roots (when they exist) sit at equal distances either side of it, so the vertex is exactly halfway between the roots.

B.3. B

Base, digit, and positional notation (Chapter 1)

The **base** (or **radix**) b of a positional system is how many times larger each position is than the one to its right. A **digit** is one of the symbols $0, 1, \ldots, b-1$. **Positional** (or **place-value**) notation means a digit's value is the digit times b^{position}, counting positions from 0 at the right. When the base is not obvious we write it as a subscript: $101101_2 = 45_{10}$.

Base, exponent, power (Chapter 3)

In a^n, a is the **base** and n is the **exponent** (also called the **index** — plural **indices** — or **power**). The whole expression a^n is "a **power** of a" and is read "a to the n", "a to the power n", or "a raised to the nth". $a^1 = a$. In code it is written `a**n` (Python), `a^n` (spreadsheets, math software), or `pow(a, n)` in C; in C and Java `^` means XOR.

Binary, bit, byte, hexadecimal (Chapter 1)

Binary is base 2, with digits 0 and 1. A single binary digit is a **bit**; eight bits make a **byte**. **Hexadecimal** ("hex") is base 16, with digits 0–9 then A=10, B=11, C=12, D=13, E=14, F=15. Because $16 = 2^4$, each hex digit corresponds to exactly four bits (a **nibble**). **Octal** is base 8 (three bits per digit). Programming languages mark bases with prefixes: `0b101101`, `0x2D`, `0o55` all mean 45.

Binomial, product (Chapter 4)

A **binomial** is an expression with two terms, such as $x + 3$ or $2a - 5b$ (*bi* = two, *nomen* = name). A **monomial** has one term, a **trinomial** three. A **product** is the result of a multiplication; "the product of $x + 3$ and $x + 2$" means $(x + 3)(x + 2)$.

Bound and free variables (Chapter 4)

A variable is **free** in an expression if the expression's value depends on it, and **bound** if it is only a placeholder inside some construction. In $\sum_{i=1}^{n} i^2$ (the sum of i^2 as i runs from 1 to n, see Chapter 14), n is free and i is bound: renaming i to k changes nothing. Programmers know this as *scope*: a loop counter is local to the loop, exactly like a bound variable.

B.4. C

Cartesian plane, axes, origin (Chapter 10)

The **Cartesian plane** (or **coordinate plane**, **xy-plane**) is the plane together with two perpendicular number lines, the **axes**. They meet at the **origin**, $O = (0, 0)$. The horizontal axis is the **x-axis**; the vertical axis is the **y-axis**.

Catastrophic cancellation (Chapter 8)

Catastrophic cancellation is the loss of significant digits when two nearly equal floating-point numbers are subtracted. The subtraction itself is exact; the problem is that the inputs' small rounding errors become large relative to the small result. The cure is almost always algebraic: rewrite the expression so the subtraction never happens.

Classifying triangles (Chapter 9)

By sides:

- **Equilateral**: all three sides equal (so all angles $60°$).
- **Isosceles**: at least two sides equal. The equal sides are the **legs**; the third is the **base**; the angles opposite the legs are the **base angles**, and they are equal.
- **Scalene**: no two sides equal.

By angles: **acute** (all angles acute), **right** (one right angle), **obtuse** (one obtuse angle). In a right triangle the side opposite the right angle is the **hypotenuse** (always the longest side) and the other two are the **legs**.

Triangle ABC is written $\triangle ABC$.

Coefficient (Chapter 4)

A **coefficient** is the numerical factor multiplying the variable part of a term. In $7x^2 - 3x + 4$, 7 is the coefficient of x^2, and -3 is the coefficient of x (the sign is part of it). In x or $-y$ the coefficients are 1 and -1, usually not written. In ax^2, with a a parameter, we also call a the coefficient.

Coefficient (Chapter 7)

A **coefficient** is the number multiplying a power of the variable. In $7x^2 - 3x + 4$, 7 is the coefficient of x^2 and -3 is the coefficient of x. A missing power has coefficient 0; a bare x^2 has coefficient 1.

Coefficients of a quadratic (Chapter 8)

In $ax^2 + bx + c$: a is the **leading** (or **quadratic**) coefficient, b the **linear** coefficient, and c the **constant term**. In $7x^2 - 3x + 4$, $a = 7$, $b = -3$, $c = 4$. Always read the signs with the coefficients: in $x^2 - 5x + 6$, $b = -5$, not 5. The condition $a \neq 0$ matters: if $a = 0$ the equation is linear, not quadratic.

Common, natural and binary logarithms (Chapter 12)

- **Common log**, base 10: $\log_{10} x$, often written $\log x$ in science and engineering and on calculators. It counts decimal digits: $\log_{10} 1000 = 3$.
- **Natural log**, base e: $\ln x = \log_e x$ ("ell-en" or "lawn"). It is the one that makes calculus clean, so mathematicians often write plain $\log x$ to mean $\ln x$.
- **Binary log**, base 2: $\log_2 x$, sometimes written $\lg x$. It counts bits and halvings, and in algorithm analysis an unadorned $\log n$ usually means $\log_2 n$.

In code: Python's `math.log(x)` is the **natural** log, and `math.log10`, `math.log2` and `math.log(x, b)` give the others. The same holds for C's `log`, JavaScript's `Math.log`, NumPy's `np.log` and Excel's `LN`. (Excel's `LOG` is base 10.) If a bare "log" shows up, check which base is meant.

Commutative, associative, distributive (Chapter 1)

An operation is **commutative** if order of the inputs does not matter: $a + b = b + a$, $ab = ba$. It is **associative** if grouping does not matter: $(a + b) + c = a + (b + c)$, $(ab)c = a(bc)$. Multiplication **distributes** over addition: $a(b + c) = ab + ac$. Mnemonic: *commute* = move around; *associate* = who you group with; *distribute* = hand out to each.

Completing the square (Chapter 8)

Completing the square means rewriting $x^2 + px$ as $\left(x + \frac{p}{2}\right)^2 - \left(\frac{p}{2}\right)^2$, so that the variable appears only once, inside a square. It is the key step in solving any quadratic and in finding its vertex (Section 8.5).

Complex conjugate (Chapter 15)

The **complex conjugate** of $z = a + bi$ is $\bar{z} = a - bi$ (also written z^*, especially in physics). Geometrically it is the reflection of z in the real axis. Key facts:

- $z\bar{z} = (a + bi)(a - bi) = a^2 - abi + abi - b^2 i^2 = a^2 + b^2 = |z|^2$ — always a non-negative *real* number.
- $z + \bar{z} = 2a = 2\,\mathrm{Re}(z)$ and $z - \bar{z} = 2bi$.
- Conjugation respects arithmetic: $\overline{z + w} = \bar{z} + \bar{w}$, $\overline{zw} = \bar{z}\,\bar{w}$.

Complex number, real part, imaginary part (Chapter 15)

A **complex number** is an expression $z = a + bi$ where a and b are real numbers. a is the **real part**, written $\mathrm{Re}(z)$; b is the **imaginary part**, written $\mathrm{Im}(z)$. Note that the imaginary part is the real number b, not bi. The set of all complex numbers is written $\mathbb{C}$. Two complex numbers are equal

exactly when their real parts are equal *and* their imaginary parts are equal. A number with $b = 0$ is real; a number with $a = 0, b \neq 0$ is called **purely imaginary**.

Composition of functions (Chapter 11)

The **composition** of f and g is the function $f \circ g$, read "f composed with g", "f after g", or "f of g":

$$(f \circ g)(x) = f(g(x))$$

Order: g is applied **first**, even though it is written second. Read $f \circ g$ from right to left, like nested function calls `f(g(x))` in code. The small circle $\circ$ is not a multiplication dot.

$f \circ g$ is defined at x only when x is in the domain of g **and** $g(x)$ is in the domain of f.

Congruence criteria (Chapter 9)

Two triangles are congruent if they match in:

- **SSS**: all three sides.
- **SAS**: two sides and the angle **between** them (the **included** angle).
- **ASA**: two angles and the side between them.
- **AAS**: two angles and a non-included side (equivalent to ASA, since the third angle is then forced by the angle sum).
- **HL** (US) / **RHS** (UK): for right triangles, the hypotenuse and one leg.

In a congruence statement $\triangle ABC \cong \triangle DEF$ the **order matters**: A corresponds to D, B to E, C to F. US textbooks then use **CPCTC**, "corresponding parts of congruent triangles are congruent", to conclude that any other matching pair of parts is equal.

Conic section (Chapter 10)

A **conic section** (**conic**) is a curve obtained by intersecting a plane with a double cone: circle, ellipse, parabola or hyperbola. Each (except the circle) can also be defined by a **focus** (a fixed point; plural **foci**) and a **directrix** (a fixed line). The **vertex** is the point where a parabola turns; the **axis of symmetry** is the line it is mirror-symmetric about.

Consistent, inconsistent, independent, dependent (Chapter 6)

- A system is **consistent** if it has at least one solution, and **inconsistent** if it has none (parallel lines).
- A consistent system is **independent** if the solution is unique (crossing lines), and **dependent** if there are infinitely many solutions (coincident lines). The second equation then "depends on" the first: it's a multiple of it and adds no new information.

US textbooks use all four words. A common summary is "consistent independent / consistent dependent / inconsistent".

Constant (Chapter 4)

A **constant** is a quantity with a fixed value throughout a discussion. Literal numbers such as 10 or $-\frac{3}{4}$ are constants, and so are named numbers such as $\pi \approx 3.14159$ and $e \approx 2.71828$.

Constant term (Chapter 4)

The **constant term** is a term with no variable in it: the 4 in $7x^2 - 3x + 4$. If there is no such term, the constant term is 0.

Constant term (Chapter 7)

The **constant term** is the term with no variable, i.e. the coefficient of x^0. In $7x^2 - 3x + 4$ it is 4. It is the value of the polynomial at $x = 0$.

Convergence of a series (Chapter 14)

An infinite series $\sum_{k=1}^{\infty} a_k$ **converges** to S if its partial sums $S_n = \sum_{k=1}^{n} a_k$ get arbitrarily close to S and stay there as $n \to \infty$ (read "as n tends to infinity"). We then write $\sum_{k=1}^{\infty} a_k = S$ and call S the **sum** of the series. If the partial sums do not approach any finite number, the series **diverges**. The symbol ∞ (**infinity**) here is not a number; $\sum^{\infty}$ is shorthand for "the limit of the partial sums".

Coordinates, ordered pair (Chapter 10)

A point's **coordinates** are the **ordered pair** (x, y). The first number, the **x-coordinate** (old name: **abscissa**), is the signed horizontal distance from the y-axis. The second, the **y-coordinate** (old name: **ordinate**), is the signed vertical distance from the x-axis. "Ordered" matters: $(2, 5)$ and $(5, 2)$ are different points.

Corner-point principle (Chapter 6)

A **linear program** (LP) maximises or minimises a linear **objective function** (like $30x + 20y$) subject to linear inequality **constraints**. The **corner-point principle**: if an LP has an optimal solution and its feasible region has at least one vertex, then some vertex is optimal. (If the objective's level lines are parallel to an edge, every point on that edge ties.)

B.5. D

Decimal places, terminating and repeating decimals (Chapter 2)

The digits after the decimal point are the **decimal places**: the first is tenths, the second hundredths, the third thousandths. A **terminating decimal** has finitely many non-zero decimal places (0.125). A **repeating** (or **recurring**) decimal ends in a block of digits that repeats forever. The repeating block is marked with a bar (US) or dots (UK): $0.1\overline{6} = 0.1666\ldots$, $0.\overline{142857} = 0.142857142857\ldots$. The length of the repeating block is the **period**.

Decimal (SI) and binary (IEC) prefixes (Chapter 3)

SI (decimal): kilo (k) $= 10^3$, mega (M) $= 10^6$, giga (G) $= 10^9$, tera (T) $= 10^{12}$, peta (P) $= 10^{15}$. **IEC (binary):** kibi (Ki) $= 2^{10}$, mebi (Mi) $= 2^{20}$, gibi (Gi) $= 2^{30}$, tebi (Ti) $= 2^{40}$, pebi (Pi) $= 2^{50}$. So 1 KB $= 1000$ bytes and 1 KiB $= 1024$ bytes. Lower-case b means bits, upper-case B bytes: 1 Gbit/s is 125 MB/s.

Degree (Chapter 7)

The **degree** of a polynomial is the highest power of the variable that appears with a non-zero coefficient. $4x^3 - 100x^2 + 600x$ has degree 3. A non-zero constant such as 5 has degree 0. Names: degree 1 **linear**, 2 **quadratic**, 3 **cubic**, 4 **quartic**, 5 **quintic**.

Degree of a term and of an expression (Chapter 4)

The **degree** of a term is the total number of variable factors in it: $7x^2$ has degree 2, $-3x$ has degree 1, 4 has degree 0, and $5x^2y$ has degree 3. The degree of a **polynomial** (a sum of such terms with whole-number powers) is the largest degree among its terms. Degree 1 means **linear**, degree 2 **quadratic**, degree 3 **cubic**. You will use these words constantly from Chapter 5 on.

Delta notation (Chapter 5)

Δ (capital Greek delta) means "change in". $\Delta x = x_2 - x_1$ is the change in x from a first point (x_1, y_1) to a second (x_2, y_2); the subscripts 1 and 2 simply label the points. $\Delta y = y_2 - y_1$. Slope is "rise over run":

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}, \qquad x_1 \neq x_2.$$

Delta notation (Chapter 10)

The Greek capital letter **delta**, Δ, is read "change in". So $\Delta x = x_2 - x_1$ is the change in x going from the first point to the second. The subscripts x_1, x_2 are just labels ("x of point 1", "x of point 2"), not powers or multiplications.

Dimensional analysis, conversion factor (Chapter 2)

Dimensional analysis (also called the **factor-label method** or **unit analysis**) converts units by multiplying by **conversion factors** — fractions equal to 1, such as $\frac{60\text{ s}}{1\text{ min}}$ — chosen so that unwanted units cancel. A **dimension** is the kind of quantity (length, time, mass); a **unit** is a particular scale for it (metre, second, kilogram). You can only add quantities with the same dimension, and both sides of a correct equation have the same dimension.

Discriminant (Chapter 8)

The **discriminant** of $ax^2 + bx + c$ is the number

$$\Delta = b^2 - 4ac.$$

(Δ is capital Greek delta; some books write D.) It "discriminates" between the cases: $\Delta > 0$, two distinct real roots; $\Delta = 0$, one repeated real root; $\Delta < 0$, no real roots. Section 8.7 explains why.

Distance formula (Chapter 10)

The **distance formula**: the distance between (x_1, y_1) and (x_2, y_2) is $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. It is the Pythagorean theorem written in coordinates. This straight-line distance is called the **Euclidean distance**.

Distance metrics (Chapter 10)

For points $p = (x_1, y_1)$ and $q = (x_2, y_2)$:

- **Euclidean** (L^2) distance: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ — the straight line.
- **Manhattan** (**taxicab**, L^1) distance: $|x_2 - x_1| + |y_2 - y_1|$ — grid moves only.
- **Chebyshev** (**chessboard**, L^∞) distance: $\max(|x_2 - x_1|, |y_2 - y_1|)$ — moves like a chess king, diagonals allowed at the same cost.

A **metric** is any distance function that is never negative, is zero only from a point to itself, is symmetric, and obeys the **triangle inequality** $d(p, r) \le d(p, q) + d(q, r)$ (a detour never helps). All three above are metrics.

Distributive law, expand (Chapter 4)

The **distributive law** says multiplication "distributes over" addition: $a(b + c) = ab + ac$. To **expand** (or "multiply out") an expression is to use this law to remove brackets, producing a sum of terms. The opposite move, rewriting a sum as a product, is **factoring** (Section 4.9).

Dividend, divisor, quotient, remainder (Chapter 7)

In $p(x) \div d(x)$, p is the **dividend** and d the **divisor**. Division produces a **quotient** $q(x)$ and **remainder** $r(x)$ such that

$$p(x) = d(x)\,q(x) + r(x), \qquad \deg r < \deg d \text{ (or } r = 0).$$

If $r = 0$, d divides p exactly, and d is a factor of p.

Divides, factor, multiple (Chapter 1)

b **divides** a, written $b \mid a$ (a vertical bar, read "b divides a"), if the remainder is 0, i.e. $a = qb$ for some integer q. Then b is a **factor** (or **divisor**) of a, and a is a **multiple** of b. So $3 \mid 12$ and $5 \nmid 12$. Do not confuse $b \mid a$ (a true/false statement) with b/a (a number).

Division with remainder (Chapter 1)

For integers a and $b > 0$ there are unique integers q (the **quotient**) and r (the **remainder**) with

$$a = qb + r, \qquad 0 \le r < b.$$

Example: $47 = 6 \times 7 + 5$, so dividing 47 by 7 gives quotient 6, remainder 5. In code, `q, r = divmod(a, b)`, or `a // b` and `a % b`.

Domain (Chapter 7)

The **domain** of an expression is the set of values of the variable for which it is defined. For a rational expression, that is every real number **except** the roots of the denominator. The domain of $\dfrac{x+1}{x^2-4}$ is all real x except $x = \pm 2$, written $x \neq \pm 2$.

Domain, codomain, range (Chapter 11)

For a function $f : A \to B$:

- The **domain** is A, the set of allowed inputs.
- The **codomain** is B, the set where outputs are declared to live (the "return type").
- The **range** (also called the **image**) is the set of outputs that actually occur: $\{f(x) : x \in A\}$. It is always a subset of the codomain, and may be smaller.

Example: $f : \mathbb{R} \to \mathbb{R},\ f(x) = x^2$. Domain $\mathbb{R}$, codomain $\mathbb{R}$, range $[0, \infty)$, since squares are never negative.

Doubling time and half-life (Chapter 12)

The **doubling time** of an exponentially growing quantity is the time it takes to double. The **half-life** of a decaying quantity is the time it takes to halve. For an exponential process these are **constant**: going from 100 to 200 takes exactly as long as going from 1,000 to 2,000.

With a half-life T, the amount after time t is

$$N(t) = N_0\left(\tfrac{1}{2}\right)^{t/T}.$$

Here N_0 ("N nought") is the starting amount and t/T counts how many half-lives have gone by.

B.6. E

Eccentricity (Chapter 10)

The **eccentricity** e of a conic is the ratio (distance to focus) / (distance to directrix), which is the same for every point on it. $e = 0$: circle. $0 < e < 1$: ellipse (for the standard ellipse, $e = c/a$). $e = 1$: parabola. $e > 1$: hyperbola. One number classifies them all.

Equation (Chapter 4)

An **equation** is a statement that two expressions are equal: *left-hand side* (LHS) = *right-hand side* (RHS). An equation may be true or false depending on the values of its variables. Example: $2(x + 3) = 14$ is true when $x = 4$ and false otherwise.

Equation of a circle (Chapter 10)

The **standard form** (or **centre–radius form**) of a circle with centre (h, k) and radius r is $(x - h)^2 + (y - k)^2 = r^2$. The circle centred at the origin is $x^2 + y^2 = r^2$; with $r = 1$ it is the **unit circle**, the star of Chapter 13. Expanding standard form gives the **general form** $x^2 + y^2 + Dx + Ey + F = 0$.

Equivalent equations (Chapter 4)

Two equations are **equivalent** if they have exactly the same solution set. $2x + 3 = 11$, $2x = 8$, and $x = 4$ are equivalent: each is true for $x = 4$ and for nothing else. Solving an equation means replacing it by a chain of equivalent equations, ending with one whose solution is obvious.

Equivalent equations (Chapter 5)

Two equations are **equivalent** if they have exactly the same solution set. $2x + 1 = 7$, $2x = 6$ and $x = 3$ are equivalent. **Solving** an equation means producing a chain of equivalent equations that ends in something so simple, like $x = 3$, that the solution set can be read off.

Equivalent systems (Chapter 6)

Two systems are **equivalent** if they have exactly the same solution set. Elimination and substitution are chains of equivalent systems, each simpler than the one before, ending in a system so simple you can read the answer off (like $x = 3,\ y = 2$).

Evaluate, substitute (Chapter 4)

To **evaluate** an expression is to compute its numerical value for given values of its variables. To **substitute** is to replace a variable by something else, a number or another expression. Evaluation is substitution followed by arithmetic.

Even and odd functions (Chapter 11)

- f is **even** if $f(-x) = f(x)$ for every x in the domain. Its graph is mirror-symmetric in the y-axis. Examples: x^2, x^4, $|x|$, $\cos x$.
- f is **odd** if $f(-x) = -f(x)$ for every x in the domain. Its graph is unchanged by a half turn (180° rotation) about the origin. Examples: x, x^3, $1/x$, $\sin x$.

Most functions are **neither**. The names come from powers: x^n is even exactly when n is even, and odd when n is odd.

Explicit formula and recursive definition (Chapter 14)

An **explicit formula** (or **closed form**, or **general term**) gives a_n directly in terms of n, for example $a_n = n^2$.

A **recursive definition** (or **recurrence relation**) gives one or more **initial conditions** and a rule expressing a_n in terms of earlier terms, for example $a_1 = 1,\ a_n = a_{n-1} + 2n - 1$. Without the initial condition a recurrence does not pin down a single sequence.

Exponential function (Chapter 12)

An **exponential function** has the variable in the **exponent**:

$$f(x) = a \cdot b^x, \qquad b > 0,\ b \neq 1.$$

- b is the **base** or **growth factor**: the number you multiply by for each unit step in x.
- a is the **initial value**, the output when $x = 0$, because $b^0 = 1$.

Compare it with a **power function** such as x^2, where the variable is in the base. x^2 and 2^x are very different functions.

Exponential versus polynomial (Chapter 3)

An **exponential** expression has the variable in the exponent: 2^n, 1.05^t, 10^x. A **polynomial** (or **power**) expression has the variable in the base: n^2, n^3, x^{10}. For any base $b > 1$ and any power k, b^n eventually exceeds n^k and the ratio grows without bound. Exponential growth with base $b > 1$ has a **doubling time**; exponential decay (base between 0 and 1) has a **half-life**.

Expression (Chapter 4)

An **expression** is a meaningful combination of numbers, variables, and operations, with no equals sign: $7x^2 - 3x + 4$, $\frac{a+b}{2}$, $2\pi r$. An expression *has a value* once its variables are given values; it does not *assert* anything.

Extraneous solution (Chapter 12)

An **extraneous solution** is a value produced by a valid-looking algebraic process that does not actually satisfy the original equation. Steps such as squaring both sides or combining logarithms can introduce them, because they can enlarge the set of allowed inputs. **Always substitute back into the original equation.**

B.7. F

Factor (Chapter 4)

The **factors** of a term (or of a number) are the pieces that are multiplied together to make it. The term $6x^2y$ has factors 6, x, x, y (and combinations of them such as $2x$ or $3xy$). Terms are *added*; factors are *multiplied*. Keeping these two words apart avoids many errors.

Factor (Chapter 7)

A **factor** of an expression is something that divides it exactly, so that the expression equals the factor times something else. $(x - 3)$ is a factor of $x^2 - 9$ because $x^2 - 9 = (x - 3)(x + 3)$. **To factor** (British: **to factorise**) means to rewrite a sum as a product.

Factor, factorise, greatest common factor (Chapter 4)

To **factor** (US) or **factorise** (UK) an expression is to rewrite it as a product. The **greatest common factor** (GCF), also called the highest common factor (HCF), of several terms is the largest expression that divides all of them: the greatest common divisor of the coefficients times each shared variable raised to its *lowest* power. The GCF of $12x^3y$, $-18x^2y^2$, and $6x^2y$ is $6x^2y$.

Factor theorem (Chapter 7)

$(x - r)$ is a factor of $p(x)$ if and only if $p(r) = 0$. ("If and only if" means both directions hold: a factor gives a zero, and a zero gives a factor.)

Factorability test (Chapter 8)

A quadratic $ax^2 + bx + c$ with integer coefficients factors into linear factors with integer (or rational) coefficients if and only if its discriminant is a perfect square. Reason: the roots $(-b \pm \sqrt{\Delta})/(2a)$ are rational exactly when $\sqrt{\Delta}$ is.

Floating point (Chapter 2)

A **floating-point number** is stored as a sign, a **significand** (or **mantissa**) of fixed length, and an **exponent**: in binary, $\pm 1.b_1 b_2 \ldots b_{52} \times 2^e$ for an IEEE 754 `double`. It is binary scientific notation (Chapter 3). Only numbers that are integers divided by a power of 2 can be represented exactly, and only if they fit in the available bits. The gap between neighbouring floats grows with their size; relative precision is roughly 16 significant decimal digits.

Floor and ceiling (Chapter 11)

- The **floor** $\lfloor x \rfloor$ is the greatest integer less than or equal to x: "round down".
- The **ceiling** $\lceil x \rceil$ is the least integer greater than or equal to x: "round up".

$\lfloor 3.7 \rfloor = 3$, $\lceil 3.2 \rceil = 4$, $\lfloor 5 \rfloor = \lceil 5 \rceil = 5$. Careful with negatives: $\lfloor -3.5 \rfloor = -4$ (down means towards $-\infty$, not towards 0), and $\lceil -3.5 \rceil = -3$.

These are **step functions**: their graphs are flat stairs with a jump at each integer.

FOIL (Chapter 7)

FOIL (First, Outer, Inner, Last) is a US mnemonic for multiplying two binomials: $(a+b)(c+d) = ac + ad + bc + bd$. It is not a separate rule, just the distributive law (the four cells of a 2-by-2 area model) with a fixed order. It does **not** extend to longer polynomials; the grid does.

Forms of a linear equation (Chapter 10)

- **Slope-intercept form:** $y = mx + b$ — slope m, **y-intercept** b (where the line crosses the y-axis).
- **Point-slope form:** $y - y_1 = m(x - x_1)$ — slope m through the known point (x_1, y_1).
- **Standard form:** $Ax + By = C$ — handles vertical lines; good for systems of equations and integer coefficients.

An **intercept** is where a graph crosses an axis: the **x-intercept** has $y = 0$, the **y-intercept** has $x = 0$.

Formula (Chapter 4)

A **formula** is an equation that expresses a rule or relationship between quantities, usually with one quantity isolated on the left, its **subject**: $A = \ell w$, $F = \frac{9}{5}C + 32$, $v = u + at$. A formula is a recipe: give values to the quantities on the right and it tells you the subject.

Function (Chapter 11)

A **function** f from a set A to a set B is a rule that assigns to **each** element of A **exactly one** element of B.

We write $f : A \to B$, read "f maps A to B" or "f from A to B". The arrow is about the whole function; it says where inputs come from and where outputs live.

Fundamental theorem of algebra (Chapter 15)

Fundamental theorem of algebra: every polynomial of degree $n \geq 1$ with complex coefficients has at least one complex root. Consequently it factors completely into n linear factors,

$$p(z) = a_n(z - z_1)(z - z_2)\cdots(z - z_n),$$

so it has exactly n roots in $\mathbb{C}$, **counted with multiplicity** (a repeated factor like $(z-2)^2$ counts twice). A number system in which every polynomial has a root is called **algebraically closed**.

B.8. G

Geometric sequence (Chapter 14)

A **geometric sequence** (or **geometric progression**, GP) multiplies by a fixed **common ratio** r each step: $a_n = r\,a_{n-1}$. Its explicit form, with first term a_1, is $a_n = a_1 r^{\,n-1}$.

Graph of a function (Chapter 11)

The **graph** of f is the set of all points $(x, f(x))$ for x in the domain. In set-builder form: $\{(x, y) : x \in \text{domain},\ y = f(x)\}$.

So "the point (a, b) is on the graph" means exactly "$f(a) = b$". Reading a graph is evaluating a function with your eyes: go along to $x = a$, then up or down to the curve, and read off the height.

Greatest common divisor and least common multiple (Chapter 2)

The **greatest common divisor** $\gcd(a, b)$ (also called greatest common factor, GCF, or highest common factor, HCF) is the largest positive integer dividing both a and b. The **least common multiple** $\operatorname{lcm}(a, b)$ is the smallest positive integer that both divide. Example: $\gcd(12, 18) = 6$, $\operatorname{lcm}(12, 18) = 36$. For positive integers, $\gcd(a, b) \cdot \operatorname{lcm}(a, b) = ab$. Two numbers with gcd 1 are **coprime** (or **relatively prime**).

Greatest common factor (GCF) (Chapter 7)

The **greatest common factor** of several terms is the largest expression dividing all of them. For $6x^3 - 9x^2$ it is $3x^2$, so $6x^3 - 9x^2 = 3x^2(2x - 3)$. Take the GCF of the coefficients and the lowest power of each variable that appears in every term.

Growth rate, decay rate (Chapter 12)

If something grows by a percentage r per step (written as a decimal, so 5% means $r = 0.05$), its growth factor is $b = 1 + r$. If it **decays** (shrinks) by r per step, then $b = 1 - r$.

- $b > 1$: **exponential growth**.
- $0 < b < 1$: **exponential decay**.

r is the **growth rate** (or decay rate). b is the **growth factor**. Don't mix the two up: a 5% rate is a factor of 1.05.

B.9. H

Half-plane, feasible region, vertex (Chapter 6)

- A **half-plane** is the set of points on one side of a line. It's **closed** (includes the boundary line, drawn solid) for $\leq$ or $\geq$, and **open** (boundary excluded, drawn dashed) for $<$ or $>$.
- The **feasible region** of a system of inequalities is the set of points satisfying every inequality. Points in it are **feasible**. If it's empty, the system is **infeasible**.
- A **vertex** (corner point) of the feasible region is a point where two boundary lines meet **and** every other constraint is satisfied. Finding each vertex means solving a 2×2 system of equations, which is where the earlier part of this chapter comes back in.

B.10. I

Identity (Chapter 4)

An **identity** is an equation that is true for *every* value of its variables, such as $2(x+3) = 2x+6$ or $(a+b)^2 = a^2 + 2ab + b^2$. Some books emphasise this with a triple bar, $2(x+3) \equiv 2x+6$, read "is identically equal to". Identities are the tools of algebra: each one licenses you to replace one expression by another.

Identity (Chapter 13)

An **identity** is an equation that holds for **every** value of the variable for which both sides are defined, such as $(a+b)^2 = a^2 + 2ab + b^2$. An ordinary **equation** such as $\sin x = \frac{1}{2}$ holds only for particular values. Some books write $\equiv$ for identities.

Identity and inverse (Chapter 1)

An **identity element** for an operation leaves things unchanged: $a + 0 = a$ (0 is the **additive identity**), $a \times 1 = a$ (1 is the **multiplicative identity**). An **inverse** undoes: $-a$ is the **additive inverse** (or **opposite**) of a because $a + (-a) = 0$; $\frac{1}{a}$ is the **multiplicative inverse** (or **reciprocal**) of $a \neq 0$ because $a \cdot \frac{1}{a} = 1$.

Identity, contradiction, conditional equation (Chapter 5)

An **identity** is an equation true for every value of the variable, such as $2(x+3) = 2x+6$. A **contradiction** is true for no value, such as $x + 1 = x + 2$. A **conditional equation** is true for some values and false for others, such as $2x = 6$. The symbol $\varnothing$ denotes the **empty set**, the set with no elements; $\mathbb{R}$ denotes the set of all real numbers.

Imaginary unit (Chapter 15)

The **imaginary unit** i is a number with the defining property $i^2 = -1$. (Electrical engineers write j, since i means current; so does Python.) The square roots of a negative number $-a$ (with $a > 0$) are $\pm i\sqrt{a}$; for example $\sqrt{-9} = 3i$.

Increasing, decreasing, monotonic (Chapter 11)

On an interval I:

- f is **increasing** if $x_1 < x_2 \Rightarrow f(x_1) < f(x_2)$ for all $x_1, x_2 \in I$ (the graph rises left to right);
- f is **decreasing** if $x_1 < x_2 \Rightarrow f(x_1) > f(x_2)$;
- f is **monotonic** on I if it is one or the other throughout.

Some books say "strictly increasing" for this and "increasing" or "non-decreasing" for the version with $\leq$. A **local maximum** is a point higher than all nearby points; a **local minimum**, lower. They occur where a function switches between increasing and decreasing.

Independent and dependent variables (Chapter 4)

When one quantity is computed from another, the input is the **independent variable** and the output is the **dependent variable**. In $y = mx + b$, x is independent, y is dependent. In a formula such as $A = \pi r^2$, r is independent and A is dependent.

Inequality symbols (Chapter 5)

$a < b$	a is **less than** b: a lies to the left of b on the number line.
$a > b$	a is **greater than** b.
$a \leq b$	a is **less than or equal to** b ("at most").
$a \geq b$	a is **greater than or equal to** b ("at least").
$a \neq b$	a is **not equal to** b.

$<$ and $>$ are **strict** inequalities; $\leq$ and $\geq$ are **non-strict** (or **weak**). The narrow end of the symbol points at the smaller number.

Injective, surjective, bijective (Chapter 11)

For $f : A \to B$:

- f is **injective** (**one-to-one**) if no two inputs share an output: $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$. "No collisions."
- f is **surjective** (**onto**) if every element of the codomain is hit: for every $y \in B$ there is an $x \in A$ with $f(x) = y$. Equivalently, range = codomain. "Nothing missed."
- f is **bijective** (a **one-to-one correspondence**) if it is both. "Perfect pairing."

A function has a two-sided inverse $f^{-1} : B \to A$ **exactly when it is bijective**.

Input, output, argument, value, image (Chapter 11)

If f sends x to y, we write $f(x) = y$, read "f of x equals y".

- x is the **input**, also called the **argument** or **independent variable**.

- y is the **output**, also called the **value of f at x**, the **image of x**, or the **dependent variable** (it depends on x).

The parentheses in $f(x)$ mean "apply f to", **not** multiplication. $f(x)$ is one object, the output.

Integer (Chapter 1)

An **integer** is a whole number that may be positive, negative, or zero: $\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots$. The set is written $\mathbb{Z}$, from the German *Zahlen*, "numbers". The integers are **closed** under addition, subtraction, and multiplication: doing any of those to two integers gives an integer.

Intercepts, zeros and roots (Chapter 11)

- The **y-intercept** is where the graph crosses the y-axis: the point $(0, f(0))$. A function has at most one.
- An **x-intercept** is where the graph meets the x-axis: a point $(a, 0)$ with $f(a) = 0$.
- A number a with $f(a) = 0$ is a **zero** of the function, also called a **root** (of the equation $f(x) = 0$). Zeros of f are the x-coordinates of its x-intercepts.

Interior and exterior angles (Chapter 9)

The **interior angles** of a polygon are the angles inside it at each vertex. Extend one side past a vertex and the angle between the extension and the next side is an **exterior angle**. Each exterior angle forms a linear pair with its interior angle.

Interval notation (Chapter 5)

An **interval** is an unbroken stretch of the number line. A **square bracket** means the endpoint is included; a **round bracket** (parenthesis) means it is excluded.

Inequality	Interval	Name
$a \le x \le b$	$[a, b]$	closed interval
$a < x < b$	(a, b)	open interval
$a \le x < b$	$[a, b)$	half-open interval
$a < x \le b$	$(a, b]$	half-open interval
$x \ge a$	$[a, \infty)$	unbounded (ray)
$x < b$	$(-\infty, b)$	unbounded (ray)
all real numbers	$(-\infty, \infty) = \mathbb{R}$	

The symbol ∞ (infinity) is not a number; $[a, \infty)$ means "no upper limit". Infinity therefore always gets a round bracket. On a number line, a filled dot marks an included endpoint and an open circle an excluded one.

Interval notation and set-builder notation (Chapter 11)

Interval notation describes runs of real numbers. A square bracket includes the endpoint; a round bracket excludes it.

- $[2, 5]$ means $2 \le x \le 5$ (closed interval).
- $(2, 5)$ means $2 < x < 5$ (open interval).
- $[2, 5)$ means $2 \le x < 5$ (half-open).
- $[0, \infty)$ means $x \ge 0$. Infinity always takes a round bracket, because it is not a number you can reach.
- $\cup$ (union) joins pieces: $(-\infty, 2) \cup (2, \infty)$ is "every real except 2", also written $\mathbb{R} \setminus \{2\}$.

Set-builder notation $\{x \in \mathbb{R} : x \neq 2\}$ reads "the set of real x such that $x \neq 2$". The colon (sometimes a vertical bar) means "such that".

Python programmers already know this: `range(2, 5)` is the half-open interval $[2, 5)$ of integers.

Inverse operation (Chapter 5)

An **inverse operation** undoes another. Adding k and subtracting k are inverses; multiplying by k and dividing by k are inverses, **provided** $k \neq 0$. Applying an operation and then its inverse returns you to where you started.

Inverse trig functions (Chapter 13)

$\sin^{-1} x$ (also $\arcsin x$), $\cos^{-1} x$ ($\arccos x$) and $\tan^{-1} x$ ($\arctan x$) return **the angle** whose sine, cosine or tangent is x.

$$\theta = \sin^{-1} x \quad \Longleftrightarrow \quad \sin\theta = x \text{ and } \theta \text{ is in the principal range.}$$

The **principal ranges** are $-90° \le \sin^{-1} x \le 90°$, $0° \le \cos^{-1} x \le 180°$ and $-90° < \tan^{-1} x < 90°$. There are infinitely many angles with a given sine, so the inverse function has to pick one (see Section 13.3).

$\sin^{-1} x$ is **not** $1/\sin x$. The -1 here means "inverse function", as in f^{-1} (Chapter 11), not "reciprocal". This is a notorious notational clash, and it's why many people prefer writing arcsin.

Irrational number (Chapter 1)

An **irrational number** is a real number that is *not* rational — it cannot be written as a ratio of integers. Its decimal expansion never terminates and never settles into a repeating block. Examples: $\sqrt{2} = 1.41421356\ldots$, $\pi = 3.14159265\ldots$, $e = 2.71828182\ldots$.

Irreducible (prime) polynomial (Chapter 7)

A polynomial is **irreducible** over a given number system if it cannot be written as a product of two polynomials of lower degree with coefficients in that system. $x^2 + 4$ is irreducible over the real numbers; $x^2 - 2$ is irreducible over the rationals but factors over the reals as $(x - \sqrt{2})(x + \sqrt{2})$. "Factor completely" means: keep going until every factor is irreducible (usually over the integers or rationals, unless stated).

B.11. K

Kinds of angle (Chapter 9)

- **Acute**: between $0°$ and $90°$.
- **Right**: exactly $90°$, marked with a small square in diagrams.
- **Obtuse**: between $90°$ and $180°$.
- **Straight**: exactly $180°$ (a straight line).
- **Reflex**: between $180°$ and $360°$.

Two lines that meet at a right angle are **perpendicular**, written $\ell_1 \perp \ell_2$. Lines in a plane that never meet are **parallel**, written $\ell_1 \parallel \ell_2$.

B.12. L

Labelling convention (Chapter 13)

In triangle ABC, the angles at the vertices are A, B, C, and the side **opposite** each angle gets the matching lower-case letter: side a faces angle A, and so on. The angles always add to $180°$.

Laws of exponents (Chapter 3)

For non-zero bases and integer exponents: $a^m a^n = a^{m+n}$, $a^m / a^n = a^{m-n}$, $(a^m)^n = a^{mn}$, $(ab)^n = a^n b^n$, $(a/b)^n = a^n / b^n$, $a^0 = 1$, $a^{-n} = 1/a^n$. They extend to fractional exponents for **positive** bases. Each can be re-derived by writing out the factors.

The laws of logarithms (Chapter 12)

For $b, c > 0$, $b, c \neq 1$, $M, N > 0$, and any real k:

Law	Logarithm form	Exponent law it comes from
Product	$\log_b(MN) = \log_b M + \log_b N$	$b^m b^n = b^{m+n}$
Quotient	$\log_b(M/N) = \log_b M - \log_b N$	$b^m / b^n = b^{m-n}$
Power	$\log_b(M^k) = k \log_b M$	$(b^m)^k = b^{mk}$
Zero	$\log_b 1 = 0$	$b^0 = 1$
Identity	$\log_b b = 1$	$b^1 = b$
Inverse	$\log_b(b^x) = x$, $b^{\log_b x} = x$	definition
Change of base	$\log_b x = \dfrac{\log_c x}{\log_c b}$	power law

Laws of sines and cosines (Chapter 13)

For any triangle:

$$\textbf{Sines:}\quad \frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C} \qquad\qquad \textbf{Cosines:}\quad a^2 = b^2 + c^2 - 2bc\cos A$$

(and the same with the letters cycled: $b^2 = a^2 + c^2 - 2ac\cos B$, $c^2 = a^2 + b^2 - 2ab\cos C$).

When $A = 90°$, $\cos A = 0$ and the law of cosines **is** Pythagoras. The term $-2bc\cos A$ is the correction for a non-right angle. It is negative (side shorter) for acute A and positive (side longer) for obtuse A.

Leading term and leading coefficient (Chapter 7)

The **leading term** is the term of highest degree; its coefficient is the **leading coefficient**. In $-2x^5 + x^2 - 9$ the leading term is $-2x^5$ and the leading coefficient is -2. A polynomial whose leading coefficient is 1 is called **monic**.

Like terms (Chapter 4)

Like terms have exactly the same variable part, with each variable raised to the same power. $5x^2y$ and $-2yx^2$ are like terms (order of multiplication does not matter). $3x$ and $3x^2$ are *not* like terms; neither are xy and x. Only like terms can be combined into a single term.

Like terms (Chapter 7)

Like terms have the same variable raised to the same power: $3x^2$ and $-5x^2$ are like terms; $3x^2$ and $3x$ are not. Only like terms can be combined: $3x^2 - 5x^2 = -2x^2$.

Linear equation (Chapter 5)

A **linear equation** in one unknown x is an equation that can be rearranged into the form

$$ax + b = 0, \qquad a \neq 0,$$

where a and b are constants. The unknown appears only to the **first power**: no x^2, no $\sqrt{x}$, no $1/x$, no x multiplied by another unknown. The name comes from the Latin *linea*, "line": graph $y = ax + b$ and you get a straight line (Section 5.10).

Linear equation (in several variables) (Chapter 6)

An equation is **linear** if each unknown appears only to the first power, multiplied by a constant, and unknowns are never multiplied together. Its **standard form** is

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b.$$

Here the subscripted $x_1, x_2, \ldots, x_n$ are the unknowns, the a_i are their **coefficients**, and b is the **constant term** (or **right-hand side**). So $3x - 2y = 7$ is linear. $xy = 7$, $x^2 + y = 1$ and $\sqrt{x} = y$ are not.

Logarithm (Chapter 12)

For a base $b > 0$, $b \neq 1$, and a positive number x:

$$\log_b x = y \quad \Longleftrightarrow \quad b^y = x.$$

Read $\log_b x$ as "log base b of x". It is **the exponent you put on b to get x**. In the equation $b^y = x$:

- b is the **base**,
- y is the **logarithm** (the exponent),
- x is sometimes called the **argument** of the log.

The double arrow $\Longleftrightarrow$ means "if and only if": each statement implies the other, so they are two ways of saying the same thing.

Logarithmic scale (Chapter 12)

A **logarithmic scale** positions a value v at distance proportional to $\log v$. Equal steps along it are equal **multiplications**. A **decade** is a factor of 10, and an **octave** (in music, and in frequency analysis) is a factor of 2.

Lowest common denominator (LCD) (Chapter 5)

The **lowest common denominator** of several fractions is the least common multiple of their denominators: the smallest positive number each denominator divides into. For $\frac{1}{3}$ and $\frac{1}{4}$ it is 12; for $\frac{1}{4}$ and $\frac{1}{6}$ it is 12, not 24. (See Chapter 2.) Multiplying an equation through by the LCD is called **clearing fractions**.

B.13. M

The "maps to" arrow and membership (Chapter 11)

$x \mapsto x^2$ (a barred arrow) reads "x maps to x^2". It describes what happens to a single input, and lets you talk about a function without naming it. Compare:

- $f : \mathbb{R} \to \mathbb{R}$ (plain arrow) says which sets are involved;
- $x \mapsto x^2$ (barred arrow) says what the rule does.

$x \in A$ reads "x is an element of A" (or "x is in A"). $\mathbb{R}$ is the set of all real numbers, $\mathbb{Z}$ the integers, $\mathbb{N}$ the natural numbers (see Chapter 1).

Matrix, augmented matrix, row echelon form, pivot (Chapter 6)

- A **matrix** is a rectangular array of numbers arranged in **rows** (horizontal) and **columns** (vertical). An $m \times n$ matrix has m rows and n columns. Matrices get a full chapter of their own in Volume 2, Chapter 27 .
- The **coefficient matrix** of a linear system holds just the coefficients. The **augmented matrix** has the right-hand-side column attached ("augmented") after a vertical bar.
- A matrix is in **row echelon form** when every row's first nonzero entry is to the right of the first nonzero entry of the row above, and any all-zero rows are at the bottom. It's a staircase ("échelon" is French for a rung of a ladder).
- The first nonzero entry in each row of an echelon form is a **pivot**. The pivot is the entry you use to clear the entries beneath it.

Midpoint formula (Chapter 10)

The **midpoint** of the segment from (x_1, y_1) to (x_2, y_2) is $M = \left(\dfrac{x_1 + x_2}{2}, \dfrac{y_1 + y_2}{2} \right)$, the coordinate-wise average. A **segment** (line segment) is the part of a line between two **endpoints**.

Modulus and argument (Chapter 15)

The **modulus** (or **absolute value**, or **magnitude**) of $z = a + bi$ is its distance from 0: $|z| = \sqrt{a^2 + b^2}$ (Pythagoras). For a real number it agrees with the ordinary absolute value.

The **argument** $\arg z$ is the angle from the positive real axis to the arrow for z, measured anticlockwise, usually in radians. It is defined only for $z \neq 0$ and only up to adding multiples of 2π. The **principal argument** is the choice in $(-\pi, \pi]$.

Monomial, binomial, trinomial (Chapter 7)

Polynomials are also named by their number of terms (after combining like terms): a **monomial** has one term ($-6x^3$), a **binomial** two ($x^2 - 9$), a **trinomial** three ($x^2 + 5x + 6$). Beyond three we just say "polynomial".

B.14. N

Natural number (Chapter 1)

The **natural numbers** are the counting numbers $0, 1, 2, 3, \ldots$. The set of them is written $\mathbb{N}$ (a "blackboard bold" N). Whether $0 \in \mathbb{N}$ is a convention that varies: computer scientists and logicians almost always include 0; some number theorists start at 1. When it matters, authors write $\mathbb{N}_0$ (with zero) or $\mathbb{Z}^+$ (positive integers, without zero). This book includes 0.

The number e (Chapter 12)

$e \approx 2.71828$ is **Euler's number**. It is irrational (its decimal expansion never repeats), like π. Two key facts about it:

- $e = \lim_{n \to \infty} (1 + 1/n)^n$, the limit of continuous compounding.
- The function e^x is **its own rate of change**. At every point, the slope of the graph of $y = e^x$ equals its height. That is why e^x is called **the** exponential function, and why it is written `exp(x)` in code.

Numerator, denominator, fraction bar (Chapter 2)

In $\frac{a}{b}$ the top number a is the **numerator** (it *numbers*, or counts, the parts) and the bottom number b is the **denominator** (it *denominates*, or names, the size of the parts: quarters, eighths). The line between them is the **fraction bar** or **vinculum**, and it means division. a/b (with a slash, or **solidus**) means the same thing. The denominator can never be 0 (Section 1.4).

B.15. O

One-to-one and inverse (Chapter 11)

A function is **one-to-one** (or **injective**) if different inputs always give different outputs: $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$. Equivalently, $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$.

If f is one-to-one with range R, its **inverse function** $f^{-1} : R \to \text{domain of } f$ is defined by

$$f^{-1}(y) = x \iff f(x) = y.$$

Read f^{-1} as "f inverse". The inverse undoes f and vice versa: $f^{-1}(f(x)) = x$ for every x in the domain of f, and $f(f^{-1}(y)) = y$ for every y in its range. In composition notation, $f^{-1} \circ f$ and $f \circ f^{-1}$ are identity functions. The domain and range swap roles.

Opposite, adjacent, hypotenuse (Chapter 13)

In a right triangle, pick one of the two acute angles and call it θ (the Greek letter **theta**, the traditional name for an angle).

- The **hypotenuse** is the longest side, across from the right angle.
- The **opposite** side is the side across from θ.
- The **adjacent** side is the side next to θ that isn't the hypotenuse.

"Opposite" and "adjacent" depend on which angle you are looking from. Swap to the other acute angle and they swap too.

Order and inequality symbols (Chapter 1)

$a < b$ means "a is less than b": a lies to the *left* of b on the number line. $a > b$ means "greater than". $a \leq b$ means "less than or equal to", $a \geq b$ "greater than or equal to", and $a \neq b$ "not equal to". The narrow end of $<$ points at the smaller number. A number is **positive** if it is > 0, **negative** if < 0, and **non-negative** if ≥ 0; zero is neither positive nor negative.

Order of operations (precedence) (Chapter 1)

Precedence says which operations are done first. The standard order is:

1. **Brackets** (parentheses) — innermost first;
2. **Exponents** (powers and roots);
3. **Multiplication and division** — equal rank, left to right;
4. **Addition and subtraction** — equal rank, left to right.

US mnemonic: **PEMDAS** ("Please Excuse My Dear Aunt Sally"). UK: **BODMAS** or **BIDMAS** (Brackets, Orders/Indices, Division, Multiplication, Addition, Subtraction). They are the same rule.

B.16. P

Parabola (Chapter 8)

The graph of $y = ax^2 + bx + c$ is a **parabola**: a symmetric U-shaped curve. It opens **upwards** (a "cup", $\cup$) if $a > 0$ and **downwards** (a "cap", $\cap$) if $a < 0$. From Greek *parabolē*, "a placing alongside"; the name comes from the ancient Greek classification of conic sections (Chapter 10).

Parallel and perpendicular slopes (Chapter 10)

Two non-vertical lines with slopes m_1, m_2 are

- **parallel** exactly when $m_1 = m_2$;
- **perpendicular** exactly when $m_1 m_2 = -1$, i.e. $m_2 = -\dfrac{1}{m_1}$: the **negative reciprocal**.

The **reciprocal** of $\frac{a}{b}$ is $\frac{b}{a}$ (flip it); the negative reciprocal flips and changes sign. Special case: horizontal ($m = 0$) and vertical lines are perpendicular, but the product rule cannot express this since a vertical slope does not exist.

Parameter (Chapter 4)

A **parameter** is a letter that is held fixed within one problem but could be changed to give a related problem. In $y = mx + b$, m and b are parameters: each choice of them gives one line from a whole family of lines. In $ax^2 + bx + c = 0$, the coefficients a, b, c are parameters.

Parameter, parametric solution (Chapter 6)

A **parameter** is a letter that stands for an arbitrary value you're free to choose, as opposed to an unknown you're trying to find (compare Chapter 4). A **parametric solution** such as $(3 - 2t,\ t)$ describes an infinite solution set by saying how every solution is built from the parameter. The unknown you set equal to the parameter is called a **free variable**.

Parts of a circle (Chapter 9)

A **circle** is the set of all points in a plane at a fixed distance, the **radius** r, from a fixed point, the **centre**.

- **Diameter** d: a chord through the centre; $d = 2r$.
- **Circumference** C: the perimeter of the circle.
- **Chord**: a segment joining two points on the circle.
- **Arc**: a piece of the circle between two points. A **minor arc** is less than half the circle, a **major arc** more, a **semicircle** exactly half.
- **Sector**: the "pizza slice" region between two radii and an arc.
- **Segment** (of a circle): the region between a chord and its arc.
- **Tangent**: a line touching the circle at exactly one point, the **point of tangency**.
- **Secant**: a line crossing the circle at two points.
- **Central angle**: an angle with its vertex at the centre. **Inscribed angle**: an angle with its vertex on the circle, its sides being chords.

Percent, percentage point (Chapter 2)

Percent means "per hundred" (Latin *per centum*): $p\% = \frac{p}{100}$. So $35\% = 0.35 = \frac{7}{20}$ and $150\% = 1.5$. A **percentage point** (pp) is the unit for the *difference* between two percentages: from 4% to 5% is a rise of 1 percentage point, which is a 25% **relative** increase.

Pi (Chapter 9)

π (Greek lowercase pi) is the ratio of any circle's circumference to its diameter:

$$\pi = \frac{C}{d} = 3.14159\,26535\ldots$$

It is the **same for every circle** because all circles are similar: scaling a circle by k multiplies both C and d by k, leaving the ratio unchanged. π is **irrational** (its decimal expansion never repeats; Chapter 1) and even **transcendental** (it is not a root of any polynomial with integer coefficients).

Piecewise function (Chapter 11)

A **piecewise function** uses different formulas on different parts of its domain:

$$f(x) = \begin{cases} x + 1 & \text{if } x < 1 \\ x - 3 & \text{if } x \geq 1 \end{cases}$$

The big brace is read "$f(x)$ equals $x + 1$ if $x < 1$, and $x - 3$ if $x \geq 1$". The conditions must not overlap (else some x would get two outputs) and together should cover the whole domain. On a graph, a **filled** dot shows an endpoint that belongs to a piece; an **open** dot shows one that does not.

Point, line, plane (Chapter 9)

A **point** is a location with no size. Points are named with capital letters: A, B, P.

A **line** is straight, has no thickness, and extends forever in both directions. The line through A and B is written $\overleftrightarrow{AB}$.

A **plane** is a flat surface with no thickness that extends forever in every direction. A sheet of paper is a finite piece of one.

Points on the same line are **collinear**; points (or lines) in the same plane are **coplanar**.

Polar coordinates (Chapter 13)

A point's **polar coordinates** (r, θ) are its distance r from the origin (the **pole**) and the angle θ from the positive x-axis (the **polar axis**), measured anticlockwise.

$$\begin{aligned} \text{Polar} \to \text{Cartesian:} &\quad x = r\cos\theta, \quad y = r\sin\theta \\ \text{Cartesian} \to \text{polar:} &\quad r = \sqrt{x^2 + y^2}, \quad \theta = \operatorname{atan2}(y, x) \end{aligned}$$

The first line is the unit-circle definition scaled by r. In the second, atan2 is the quadrant-aware inverse tangent (Section 13.9). Polar coordinates aren't unique: (r, θ), $(r, \theta + 2\pi)$ and $(-r, \theta + \pi)$ all name the same point.

Polar form (Chapter 15)

The **polar form** (or **trigonometric form**, or **modulus–argument form**) of a nonzero complex number is $z = r(\cos\theta + i\sin\theta)$, where $r = |z|$ and $\theta = \arg z$. It is often abbreviated $r\,\mathrm{cis}\,\theta$ ("r cis theta": **c**os + **i** **s**in), and, after the next section, written $re^{i\theta}$. Conversion: $a = r\cos\theta$, $b = r\sin\theta$; $r = \sqrt{a^2 + b^2}$, $\theta = \operatorname{atan2}(b, a)$.

Polygon vocabulary (Chapter 9)

A **polygon** is a closed plane figure made of straight segments (**sides** or **edges**) meeting at **vertices**. It is **simple** if its sides don't cross each other, and **convex** if every interior angle is less than $180°$ (equivalently, the segment joining any two interior points stays inside). A polygon with a "dent" is **concave**. A **diagonal** joins two non-adjacent vertices.

A **regular** polygon has all sides equal **and** all angles equal. Names by number of sides: triangle (3), quadrilateral (4), pentagon (5), hexagon (6), heptagon (7), octagon (8), decagon (10), n-gon in general.

Perimeter is the total length of the boundary. **Area** is the amount of surface enclosed, measured in square units.

Polynomial (Chapter 7)

A **polynomial** in x is an expression of the form

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

where n is a non-negative whole number and the a_k are constants (the coefficients). From Greek *poly* (many) plus Latin *nomen* (name, term): "many terms".

Prime and composite (Chapter 1)

A **prime** is a natural number greater than 1 whose only positive divisors are 1 and itself: $2, 3, 5, 7, 11, 13, \ldots$. A natural number greater than 1 that is not prime is **composite**: it can be written as a product of two smaller natural numbers. 1 is neither prime nor composite. 2 is the only even prime.

Product notation and the empty product (Chapter 14)

$\prod_{i=m}^{n} f(i) = f(m) \cdot f(m+1) \cdots f(n)$. An **empty product** (upper bound below lower bound) equals 1, the identity for multiplication, just as an empty sum equals 0, the identity for addition. This is why $0! = 1$. In code: `total = 0` before a summing loop, `prod = 1` before a multiplying loop.

Proof vocabulary (Chapter 9)

- **Given**: the hypotheses you may assume.
- **Prove**: the conclusion you must reach.
- A **conditional** statement has the form "if P, then Q" ($P \Rightarrow Q$). P is the **hypothesis**, Q the **conclusion**.
- The **converse** of "if P then Q" is "if Q then P". It may or may not be true: "if a quadrilateral is a square, it has four equal sides" is true, but its converse is false (a rhombus).
- A **counterexample** is a single case showing a general statement is false.
- **Deductive** reasoning goes from general rules to specific conclusions (proof). **Inductive** reasoning (in the everyday sense, not mathematical induction) generalises from examples (conjecture).

Volume 3, Chapter 21 and Volume 3, Chapter 22 treat all of this formally.

Proper, improper, unit fraction, mixed number (Chapter 2)

A **proper fraction** has numerator smaller than denominator in size ($\frac{3}{4}$); its value lies between -1 and 1. An **improper fraction** has numerator at least as large as denominator ($\frac{7}{4}$). A **unit fraction** has numerator 1 ($\frac{1}{4}$). A **mixed number** writes an improper fraction as a whole number plus a proper fraction: $1\frac{3}{4} = 1 + \frac{3}{4} = \frac{7}{4}$. "Improper" is not an insult; in algebra improper fractions are usually *preferred*.

Pure function and side effect (Chapter 11)

A programming function is **pure** if

1. its output depends **only** on its arguments (no reading of global state, clocks, random numbers, files or networks), and
2. it has no **side effects**: it changes nothing outside itself (no writing to globals, files, screens or its arguments).

A pure function is a mathematical function. Calling it twice with the same arguments must give the same result, so the call can always be replaced by its value (**referential transparency**).

Pythagorean theorem (Chapter 9)

In a right triangle with legs a and b and hypotenuse c,

$$a^2 + b^2 = c^2.$$

Converse: if a triangle's sides satisfy $a^2 + b^2 = c^2$, the angle opposite c is a right angle.

A **Pythagorean triple** is a set of positive whole numbers satisfying the equation: (3, 4, 5), (5, 12, 13), (8, 15, 17), (7, 24, 25), and any multiple such as (6, 8, 10).

B.17. Q

Quadrant (Chapter 10)

The axes cut the plane into four **quadrants**, numbered with Roman numerals **anticlockwise** starting from the top right: **I** (x positive, y positive), **II** (x negative, y positive), **III** (both negative), **IV** (x positive, y negative). Points that lie **on** an axis belong to no quadrant.

Quadratic expression and quadratic equation (Chapter 8)

A **quadratic expression** is a polynomial of degree 2: $ax^2 + bx + c$ with $a \neq 0$. A **quadratic equation** is an equation that can be rearranged to

$$ax^2 + bx + c = 0, \qquad a \neq 0,$$

its **standard form**. The word comes from Latin *quadratum*, "a square", because the defining feature is the squared unknown x^2. (Confusingly, *quad* usually means four; here it refers to the four sides of a square.)

Quadratic formula (Chapter 8)

The solutions of $ax^2 + bx + c = 0$ ($a \neq 0$) are

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

Read it as "the axis of symmetry $-b/(2a)$, plus or minus a half-width $\sqrt{\Delta}/(2a)$". The two roots are symmetric about the vertex.

Quadrilaterals (Chapter 9)

- **Parallelogram**: both pairs of opposite sides parallel (and therefore equal).

- **Rectangle**: a parallelogram with four right angles.
- **Rhombus**: a parallelogram with four equal sides.
- **Square**: both a rectangle and a rhombus.
- **Trapezoid** (US) / **trapezium** (UK): exactly one pair of parallel sides (some US texts say "at least one"). The parallel sides are the **bases**.
- **Kite**: two pairs of adjacent equal sides.

Watch the transatlantic trap: in the UK a **trapezium** has one pair of parallel sides; in the US a **trapezium** traditionally has *none*.

B.18. R

Radian (Chapter 13)

An angle of θ **radians** at the centre of a circle of radius r cuts off an arc of length $s = r\theta$. Radians are a ratio of two lengths, so they have no physical unit, and "rad" is often left out. **If no unit is written, assume radians.**

Converting: multiply degrees by $\dfrac{\pi}{180}$ to get radians, and multiply radians by $\dfrac{180}{\pi}$ to get degrees.

Common values: $30° = \frac{\pi}{6}$, $45° = \frac{\pi}{4}$, $60° = \frac{\pi}{3}$, $90° = \frac{\pi}{2}$, $180° = \pi$, $270° = \frac{3\pi}{2}$, $360° = 2\pi$.

Ratio, rate, unit rate, proportion (Chapter 2)

A **ratio** compares quantities: $a : b$ (read "a to b") means "for every a of the first, there are b of the second", and it carries the same information as the fraction $\frac{a}{b}$. A **rate** is a ratio of quantities with different units: 60 km per hour, \$3 per kilogram. A **unit rate** has 1 in the second place: "per one". A **proportion** is an equation between two ratios, $\frac{a}{b} = \frac{c}{d}$. Two quantities are **directly proportional**, written $y \propto x$, if $y = kx$ for a fixed number k, the **constant of proportionality**.

Rational exponents (Chapter 3)

For $a > 0$ and integers m, n with $n > 0$: $a^{1/n} = \sqrt[n]{a}$ and $a^{m/n} = (\sqrt[n]{a})^m$. All the laws of exponents continue to hold. So $\sqrt{x} = x^{1/2}$, $\dfrac{1}{\sqrt{x}} = x^{-1/2}$, $\sqrt[3]{x^2} = x^{2/3}$. Converting roots to fractional exponents is how calculus handles them (Volume 2, Chapter 17).

Rational expression (Chapter 7)

A **rational expression** is a quotient of two polynomials, $\dfrac{P(x)}{Q(x)}$ with Q not the zero polynomial. "Rational" here means "ratio", as in rational numbers p/q.

Rational function and asymptote (Chapter 11)

A **rational function** is a quotient of two polynomials, $R(x) = \dfrac{P(x)}{Q(x)}$, with Q not the zero polynomial (the name comes from **ratio**, as with rational numbers). Its natural domain is every real number except the zeros of Q.

An **asymptote** is a line that the graph approaches ever more closely:

- a **vertical asymptote** $x = a$: as x gets near a, $R(x)$ grows without bound (towards $+\infty$ or $-\infty$);
- a **horizontal asymptote** $y = L$: as $x \to \infty$ or $x \to -\infty$, $R(x)$ gets as close to L as you like.

"$x \to \infty$" reads "as x grows without bound". The precise language for "gets as close as you like" is the **limit**, defined in Volume 2, Chapter 16.

Rational number (Chapter 1)

A **rational number** is any number that can be written as a ratio $\dfrac{p}{q}$ of integers p and q with $q \neq 0$. The set is written $\mathbb{Q}$, for *quotient*. Examples: $\frac{1}{2}$, $-\frac{7}{3}$, $5 = \frac{5}{1}$, $0.125 = \frac{1}{8}$. In decimal form a rational number either terminates or eventually repeats (shown in Chapter 2).

Rational root theorem (Chapter 7)

If a polynomial with integer coefficients has a rational root p/q in lowest terms, then p divides the constant term a_0 and q divides the leading coefficient a_n. For a **monic** polynomial ($a_n = 1$), every rational root is an integer dividing a_0.

Real number (Chapter 1)

The **real numbers**, written $\mathbb{R}$, are all the rational and irrational numbers together: every point on the number line. Every length, every measurement, every point on a continuous scale is a real number. The defining extra property of $\mathbb{R}$ over $\mathbb{Q}$ is that it has *no gaps* (mathematicians call this **completeness**).

Reciprocal (Chapter 2)

The **reciprocal** of a non-zero number x is $\frac{1}{x}$, the number you multiply x by to get 1. The reciprocal of $\frac{a}{b}$ is $\frac{b}{a}$ ("turn it upside down"), because $\frac{a}{b} \cdot \frac{b}{a} = \frac{ab}{ba} = 1$. Zero has no reciprocal.

Reciprocal ratios (Chapter 13)

Three more names you'll meet in older texts and in calculus:

$$\csc\theta = \frac{1}{\sin\theta}\ \text{(cosecant)}, \qquad \sec\theta = \frac{1}{\cos\theta}\ \text{(secant)}, \qquad \cot\theta = \frac{1}{\tan\theta}\ \text{(cotangent)}.$$

Note the pairing: *co*secant goes with sine, and secant goes with *co*sine. They add no new information, only convenient shorthand.

Recurrence relation and order (Chapter 14)

A **recurrence relation** defines each term of a sequence from earlier terms. Its **order** is how far back it reaches: $a_n = 3a_{n-1}$ is first order, $F_n = F_{n-1} + F_{n-2}$ is second order. An order-k recurrence needs k **initial conditions**. **Solving** a recurrence means finding an explicit formula for a_n.

Remainder theorem (Chapter 7)

The remainder when $p(x)$ is divided by $(x - r)$ is $p(r)$.

Root and zero (Chapter 7)

A **root** of the equation $p(x) = 0$, also called a **zero** of the polynomial p, is a number r with $p(r) = 0$. On the graph $y = p(x)$, real roots are the x-coordinates where the curve meets the x-axis (the **x-intercepts**). "Root", "zero" and "solution" are used almost interchangeably; strictly, equations have **solutions** or **roots** and functions have **zeros**.

Root, zero, solution (Chapter 8)

A **root** or **solution** of $ax^2 + bx + c = 0$ is a value of x that makes the equation true. The same number is called a **zero** of the function $f(x) = ax^2 + bx + c$, because f takes the value zero there. On a graph, real roots are the **x-intercepts**: where the curve meets the x-axis. A quadratic has at most two roots.

Roots of unity (Chapter 15)

The **nth roots of unity** are the n complex solutions of $z^n = 1$: $\omega^k = e^{2\pi ik/n}$ for $k = 0, 1, \ldots, n-1$, where $\omega = e^{2\pi i/n}$ is the **primitive** nth root of unity (ω is the Greek letter omega). They lie on the unit circle at the vertices of a regular n-gon, one vertex at 1. For $n \geq 2$ they sum to zero.

Rounding, significant figures, truncation (Chapter 1)

To **round** a number is to replace it by a nearby number with fewer digits. Rounding "to the nearest 10", "to 2 decimal places" (2 d.p.), or "to 3 significant figures" (3 s.f.) specifies which digit is the last one kept. The standard rule: look at the first digit dropped; if it is 5 or more round up, otherwise round down. **Truncation** simply drops digits (always toward zero). The symbol $\approx$ means "approximately equal to".

B.19. S

Scientific notation (Chapter 3)

A number is in **scientific notation** (UK: **standard form**) when written as $m \times 10^k$ with $1 \leq |m| < 10$ and k an integer. m is the **significand** (or **mantissa** or **coefficient**). Programming languages and calculators write 5.97×10^{24} as `5.97e24` (**E notation**). **Engineering notation** restricts k to multiples of 3, matching the prefixes kilo, mega, milli, micro.

Screen coordinates and pixels (Chapter 10)

Screen (or **device**, **pixel**) **coordinates** place the origin at the top-left, with x (column) rightward and y (row) downward. A **pixel** ("picture element") is one cell of the grid. **World coordinates** are the mathematical coordinates of the scene being drawn. A **viewport transform** converts world to

screen coordinates.

Segment, ray, length, midpoint (Chapter 9)

A **line segment** $\overline{AB}$ is the part of a line between endpoints A and B. Its **length** is written AB (no bar). A **ray** $\overrightarrow{AB}$ starts at A and continues forever through B.

The **midpoint** of $\overline{AB}$ is the point M on it with $AM = MB$. Two segments with equal lengths are **congruent**, written $\overline{AB} \cong \overline{CD}$.

Sequence, term, index (Chapter 14)

A **sequence** is an ordered list of numbers $a_1, a_2, a_3, \ldots$. Each number is a **term**. The subscript is the **index** (position): a_n is read "a sub n" and means "the term in position n". The whole sequence is written (a_n) or $\{a_n\}_{n \geq 1}$. A sequence may be **finite** (it stops) or **infinite** (it goes on forever). Formally, an infinite sequence is just a function whose domain is the positive integers: a_n is another way of writing $a(n)$.

Series and partial sum (Chapter 14)

A **series** is a sum of the terms of a sequence. The **nth partial sum** is $S_n = \sum_{k=1}^{n} a_k$. An **arithmetic series** sums an arithmetic sequence; a **geometric series** sums a geometric one. The partial sums $S_1, S_2, S_3, \ldots$ themselves form a new sequence — this observation is the key to infinite series.

Set-builder notation, membership, union and intersection (Chapter 5)

Set-builder notation $\{x \mid x > 2\}$ is read "the set of all x such that $x > 2$". The vertical bar means "such that" (some books use a colon). $x \in S$ means "x is an element of the set S"; for example $3 \in (2, \infty)$. The **union** $A \cup B$ is everything in A *or* B (or both); the **intersection** $A \cap B$ is everything in *both*. These are developed properly in Volume 3, Chapter 21 .

Sigma notation (Chapter 14)

$\sum_{i=m}^{n} f(i) = f(m) + f(m+1) + \cdots + f(n)$. The **index** i is a dummy variable; m and n are the **lower** and **upper bounds** (inclusive); $f(i)$ is the **summand**. If $n < m$ the sum is **empty** and equals 0. The number of terms is $n - m + 1$.

Significant figures (Chapter 3)

The **significant figures** (s.f., or **significant digits**) of a measured number are the digits that carry real information about its precision. Rules: non-zero digits are significant; zeros *between* non-zero digits are significant (4007: 4 s.f.); leading zeros are not (0.0072: 2 s.f.); trailing zeros after a decimal point are (2.50: 3 s.f.); trailing zeros in a whole number are ambiguous (1500: 2, 3, or 4?) — scientific notation removes the ambiguity: 1.5×10^3 vs 1.500×10^3.

Similar figures and scale factor (Chapter 9)

$\triangle ABC \sim \triangle DEF$ ("is similar to") means corresponding angles are equal and corresponding sides are in a common ratio:

$$\frac{AB}{DE} = \frac{BC}{EF} = \frac{CA}{FD}$$

$$\frac{DE}{AB} = \frac{EF}{BC} = \frac{FD}{CA} = k,$$

where k is the **scale factor** from the first triangle to the second. $k > 1$ is an enlargement, $0 < k < 1$ a reduction. Congruence is the special case $k = 1$.

A **proportion** is an equation stating two ratios are equal, such as $\frac{a}{b} = \frac{c}{d}$. Its solution method, **cross-multiplying**, gives $ad = bc$.

Similarity criteria for triangles (Chapter 9)

- **AA**: two pairs of equal angles (the third pair then follows from the angle sum).
- **SSS~**: all three pairs of sides in the same ratio.
- **SAS~**: two pairs of sides in the same ratio, and the included angles equal.

Simple harmonic motion (Chapter 13)

Simple harmonic motion is oscillation described by $x(t) = A\cos(\omega t + \varphi)$.

- A: **amplitude** (maximum displacement).
- ω: **angular frequency**, in radians per second.
- $T = \dfrac{2\pi}{\omega}$: **period**, in seconds per cycle.
- $f = \dfrac{1}{T} = \dfrac{\omega}{2\pi}$: **frequency**, in cycles per second (hertz, Hz).
- φ: **phase constant** (where in the cycle the motion starts).

The period of SHM doesn't depend on the amplitude. A small bounce and a big bounce take the same time.

Simplify (Chapter 4)

To **simplify** an expression is to rewrite it in an equivalent form that is shorter or more useful, typically with brackets removed, like terms collected, and terms written in a standard order (usually decreasing powers: $2x^2 + 7x - 4$). "Simplest" depends on purpose: sometimes the factored form is the more useful one.

Sine, cosine, tangent (Chapter 13)

For an acute angle θ in a right triangle:

$$\sin\theta = \frac{\text{opposite}}{\text{hypotenuse}}, \qquad \cos\theta = \frac{\text{adjacent}}{\text{hypotenuse}}, \qquad \tan\theta = \frac{\text{opposite}}{\text{adjacent}}.$$

The mnemonic is **SOH-CAH-TOA**: **S**ine = **O**pposite/**H**ypotenuse, **C**osine = **A**djacent/**H**ypotenuse, **T**angent = **O**pposite/**A**djacent.

Dividing the first by the second gives $\tan\theta = \dfrac{\sin\theta}{\cos\theta}$.

Notation: $\sin\theta$ means "sine of θ". The brackets are optional for a single symbol but needed for anything longer: $\sin(2\theta + 1)$. $\sin^2\theta$ means $(\sin\theta)^2$, **not** $\sin(\sin\theta)$.

Sinusoid parameters (Chapter 13)

For $y = A\sin(B(x - C)) + D$ (with $A, B > 0$):

- **Amplitude** A: the height of a peak above the midline, half the peak-to-trough distance. (Vertical stretch.)
- **Period** $\dfrac{2\pi}{B}$: the length of one full cycle. (A horizontal squeeze by factor B.) The **frequency** is $\dfrac{B}{2\pi}$ cycles per unit, the reciprocal of the period. B itself is the **angular frequency**, usually written ω (omega), in radians per unit.
- **Phase shift** C: the graph moves right by C. (Horizontal translation.)
- **Midline** $y = D$: the level the wave oscillates about. (Vertical translation.)

Slope (Chapter 10)

The **slope** (or **gradient**) m of a non-vertical line is $\frac{\Delta y}{\Delta x}$ between any two of its points — the change in y per unit increase in x. Positive slope rises left-to-right, negative slope falls, zero slope is horizontal, and a vertical line has **undefined** slope ($\Delta x = 0$, so you would divide by zero).

Slope, gradient, intercept (Chapter 5)

In $y = mx + b$ (the **slope–intercept form**):

- m is the **slope** (in UK usage, the **gradient**): the change in y per unit change in x. Positive slope rises to the right; negative slope falls; zero slope is horizontal.
- b is the **y-intercept**: the line crosses the y-axis at $(0, b)$. (UK texts often write $y = mx + c$.)
- The **x-intercept** is where the line crosses the x-axis, found by solving $mx + b = 0$, so $x = -b/m$ when $m \neq 0$.

The slope has **units**: units of y per unit of x.

Solids (Chapter 9)

- A **prism** has two congruent parallel polygon **bases** joined by rectangular (or parallelogram) faces. A box is a rectangular prism; a cube has six square faces.
- A **cylinder** is the circular version of a prism.
- A **pyramid** has a polygon base and triangular faces meeting at a point, the **apex**. A **cone** is the circular version.
- A **sphere** is the set of points in space at distance r from a centre.
- **Volume** is the space enclosed, in cubic units. **Surface area** is the total area of all the faces or curved surfaces, in square units. **Lateral** surface area excludes the bases.

- The **height** h is always perpendicular to the base. A cone's **slant height** ℓ is measured along the surface from apex to rim: $\ell = \sqrt{r^2 + h^2}$.

Solution of a system (Chapter 6)

A **solution** of a system is an assignment of values to **all** the unknowns that makes **every** equation true. For two unknowns x, y a solution is an **ordered pair** (x, y); for three it's an **ordered triple** (x, y, z). The **solution set** is the set of all solutions. It may contain exactly one element, none, or infinitely many.

Solution, root, solution set (Chapter 4)

A **solution** of an equation is a value of the unknown that makes it true. For an equation of the form "expression $= 0$", a solution is also called a **root**. The **solution set** is the collection of all solutions, written with braces: the solution set of $x^2 = 9$ is $\{-3, 3\}$. An equation may have one solution, several, infinitely many, or none (solution set $\varnothing$, the empty set).

Solution, root, solution set (Chapter 5)

A **solution** of an equation is a value of the unknown that makes the equation a true statement. For equations of the form $f(x) = 0$, a solution is also called a **root**. The **solution set** is the collection of all solutions. For a linear equation it contains exactly one number, no numbers, or every number (Section 5.4). For an inequality it is usually a whole interval.

Special products (Chapter 7)

$$\begin{aligned} (a \pm b)^2 &= a^2 \pm 2ab + b^2 \\ (a + b)(a - b) &= a^2 - b^2 \\ a^3 \pm b^3 &= (a \pm b)(a^2 \mp ab + b^2) \end{aligned}$$

The $\pm$ ("plus or minus") and $\mp$ ("minus or plus") signs mean: read all top signs together, or all bottom signs together.

Square root, cube root, nth root (Chapter 3)

A **square root** of a is a number whose square is a. Every positive number has two (3 and -3 are both square roots of 9). The symbol $\sqrt{a}$ — the **radical sign** over the **radicand** a — means the **principal** (non-negative) square root: $\sqrt{9} = 3$. A **cube root** $\sqrt[3]{a}$ satisfies $(\sqrt[3]{a})^3 = a$; every real number has exactly one real cube root ($\sqrt[3]{-8} = -2$). In general $\sqrt[n]{a}$ is the **nth root**; the small n is the **index** of the root.

Squared, cubed (Chapter 3)

a^2 is read "a **squared**" because it is the area of a square with side a; a^3 is "a **cubed**", the volume of a cube with edge a. A **perfect square** is the square of an integer (1, 4, 9, 16, 25, ...); a **perfect cube** is the cube of one (1, 8, 27, 64, ...).

Standard form (Chapter 7)

A polynomial is in **standard form** when like terms are combined and the terms are written in decreasing order of degree: $3x^4 - x^2 + 5x - 8$, not $5x - 8 + 3x^4 - x^2$.

Subject of a formula, inverse operation (Chapter 4)

The **subject** of a formula is the variable standing alone on one side, usually the left: F is the subject of $F = \frac{9}{5}C + 32$. **Changing the subject** (also **rearranging**, **transposing**, or **solving for** a variable) means producing an equivalent formula with a different subject. An **inverse operation** undoes another: addition and subtraction are inverses, as are multiplication and division (by a non-zero number), and squaring and taking the square root (with care about signs).

Sum, difference, product, quotient (Chapter 1)

The result of an addition is a **sum**; its inputs are **terms** (or **addends**). The result of a subtraction is a **difference**. The result of a multiplication is a **product**; its inputs are **factors**. The result of a division is a **quotient**: in $a \div b$, a is the **dividend** and b the **divisor**. Multiplication may be written $a \times b$, $a \cdot b$, or simply ab when at least one factor is a letter.

Surd, radical form (Chapter 3)

A **surd** (UK usage) or **radical** (US usage) is an irrational root such as $\sqrt{2}$ or $\sqrt[3]{5}$, left in exact form rather than as a decimal. **Simplest radical form** means: no perfect-square factor under a square root, no root in a denominator, and no fraction under a root. **Like surds** have the same radicand ($3\sqrt{2}$ and $5\sqrt{2}$) and can be added like like terms.

System of equations (Chapter 6)

A **system of equations** is a collection of equations that must all hold **at the same time**, for the same values of the unknowns. A brace on the left is the usual way to show they belong together:

$$\begin{cases} a + c = 50 \\ 12a + 7c = 465 \end{cases}$$

British courses call these **simultaneous equations**, a name that stresses the "at the same time" part. US courses say **systems of equations**. They mean the same thing.

B.20. T

Term (Chapter 4)

The **terms** of an expression are the pieces that are added together. In $7x^2 - 3x + 4$ the terms are $7x^2$, $-3x$, and 4. Subtraction is read as adding a negative, so the minus sign travels with the term that follows it. Inside brackets things are grouped: $3(x + 2) - 4y$ has two terms at the top level, $3(x + 2)$ and $-4y$.

Term (Chapter 7)

A **term** is one of the pieces being added in an expression. In $7x^2 - 3x + 4$ the terms are $7x^2$, $-3x$ and 4. The sign belongs to the term: the middle term is $-3x$, not $3x$.

Three-dimensional coordinates (Chapter 10)

3D Cartesian coordinates (x, y, z) locate a point in space by signed distances along three mutually perpendicular axes. The system is **right-handed** when x, y, z follow the right-hand rule (the standard convention in mathematics and physics). Space, thought of as all such triples, is written $\mathbb{R}^3$ ("R three"); the plane is $\mathbb{R}^2$.

Transformation, image, isometry (Chapter 10)

A **transformation** of the plane is a rule that sends every point to a new point. The **image** of A is written A' ("A prime"). An **isometry** or **rigid motion** preserves all distances (and hence angles and areas): translations, reflections and rotations are isometries, and they produce **congruent** figures. A **dilation** scales all distances by the same factor and produces **similar** figures.

Triangular numbers (Chapter 14)

The numbers $T_n = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$ (1, 3, 6, 10, 15, ...) are the **triangular numbers**, because that many dots can be arranged in a triangle. Note $T_n = \binom{n+1}{2}$, the number of ways to choose 2 items from $n + 1$ (Volume 3, Chapter 24).

Two's complement, overflow, signed and unsigned (Chapter 1)

An **unsigned** n-bit integer represents 0 to $2^n - 1$. A **signed** n-bit integer in **two's complement** represents -2^{n-1} to $2^{n-1} - 1$; the top bit is the **sign bit** (1 = negative). **Overflow** is when the true mathematical result of an operation lies outside the representable range, so the stored result wraps around. For 32 bits the signed range is $-2\,147\,483\,648$ to $2\,147\,483\,647$.

B.21. U

Unit-circle definition of sine and cosine (Chapter 13)

Rotate the point $(1, 0)$ anticlockwise about the origin by angle θ (clockwise if θ is negative). The point lands at P. Then

$$\cos\theta = x\text{-coordinate of } P, \qquad \sin\theta = y\text{-coordinate of } P, \qquad \tan\theta = \frac{\sin\theta}{\cos\theta} = \text{slope of } OP.$$

For an angle θ in standard position, the **initial side** is the positive x-axis and the **terminal side** is the ray OP. The four **quadrants** are numbered I to IV anticlockwise, starting from the top right.

Unknown (Chapter 4)

An **unknown** is a variable whose value we are trying to find from an equation. In $3x + 4 = 19$, x is the unknown, and the equation pins it down to $x = 5$. "Unknown" describes a variable's job in a problem rather than a different kind of object.

B.22. V

Variable (Chapter 4)

A **variable** is a symbol, usually a letter, that stands for a number we either do not know yet or want to leave unspecified. In $2n + 10$, n is the variable. The word comes from Latin *variabilis*, "changeable".

Vertex (Chapter 8)

The **vertex** of a parabola is its turning point: the lowest point if $a > 0$ (a **minimum**), the highest if $a < 0$ (a **maximum**). For $y = ax^2 + bx + c$ it is at $x = -\dfrac{b}{2a}$. The y-value there is the **minimum value** or **maximum value** of the quadratic.

Vieta's formulas (Chapter 8)

For $ax^2 + bx + c = 0$ with roots r_1, r_2 (real or complex, counted with multiplicity):

$$r_1 + r_2 = -\frac{b}{a}, \qquad r_1 r_2 = \frac{c}{a}.$$

For a monic quadratic $x^2 + bx + c$: the roots **sum** to $-b$ and **multiply** to c. This is exactly why factoring $x^2 + bx + c$ means "find two numbers with sum b and product c" (the numbers are the **negatives** of the roots).

B.23. Z

Zero and negative exponents (Chapter 3)

For any $a \neq 0$: $a^0 = 1$ (the empty product), and $a^{-n} = \dfrac{1}{a^n}$. A negative exponent means "reciprocal", not "negative": $2^{-3} = \frac{1}{8}$, which is positive. A negative exponent on a fraction flips it: $\left(\frac{a}{b}\right)^{-n} = \left(\frac{b}{a}\right)^{n}$.

Zero-product property (Chapter 7)

If $AB = 0$ then $A = 0$ or $B = 0$ (or both). This is why factoring solves equations. It works **only** with zero on the right: $AB = 6$ tells you nothing about A or B individually.

Appendix C: Notation and Symbols

Mathematical notation is a compressed language. This appendix is its phrasebook: for each symbol, how to say it aloud, what it means, a tiny example, and the chapter where it is properly introduced. Saying a formula aloud is half of understanding it, so the "Read aloud" column matters more than it looks.

Where a symbol has more than one common meaning, all are listed; context decides. The section Section C.13 collects the worst offenders, and Section C.14 shows how to take a dense formula apart.

C.1. Arithmetic, comparison, and approximation

Symbol	Read aloud	Meaning	Example	Ch.
$=$	"equals"	Both sides name the same value	$2 + 3 = 5$	Chapter 1
$\neq$	"is not equal to"	The two sides differ	$0.1 + 0.2 \neq 0.3$ in floating point	Chapter 1
$<$, $>$	"is less than", "is greater than"	Strict order; the small end points at the smaller number	$-3 < 2$	Chapter 1
$\leq$, $\geq$	"is less than or equal to", "is greater than or equal to"	Order allowing equality	$x \geq 0$	Chapter 5
$\approx$	"is approximately equal to"	Close in value; how close depends on context	$\pi \approx 3.1416$	Chapter 1
$\equiv$	"is identically equal to"; "is congruent to"; "is logically equivalent to"	(1) true for every value of the variable; (2) same remainder modulo m; (3) same truth value in every case	$(x+1)^2 \equiv x^2 + 2x + 1$; $17 \equiv 2 \pmod 5$	Chapter 4, Volume 3, Chapter 23 , Volume 3, Chapter 21
$:=$	"is defined to be"	Defines the left side; not a claim to check	$f(x) := x^2$	Chapter 4
$\propto$	"is proportional to"	Equal up to a constant multiplier	$A \propto r^2$ for circles (constant π)	Chapter 2
$\ll$, $\gg$	"is much less than", "is much greater than"	Smaller (larger) by orders of magnitude	$10^{-16} \ll 1$	Volume 2, Chapter 32

Symbol	Read aloud	Meaning	Example	Ch.
$\pm$, $\mp$	"plus or minus", "minus or plus"	Two cases at once; $\mp$ takes the opposite sign to a $\pm$ in the same formula	$x = 3 \pm 2$ means $x = 5$ or $x = 1$	Chapter 8
$\times$, $\cdot$	"times"	Multiplication (also cross product and dot product for vectors, Cartesian product for sets)	$3 \times 4 = 3 \cdot 4 = 12$	Chapter 1
a/b, $\frac{a}{b}$, $a \div b$	"a over b", "a divided by b"	Division; a fraction	$\frac{6}{4} = 1.5$	Chapter 2
$a : b$	"a to b"	Ratio	$3 : 2$	Chapter 2
$\%$	"percent"	Per hundred: $p\% = p/100$	$15\% = 0.15$	Chapter 2
a^n	"a to the n", "a to the power n" (a^2 "a squared", a^3 "a cubed")	Repeated multiplication, extended to all real n	$2^{-1} = \frac{1}{2}$	Chapter 3
$\sqrt{a}$, $\sqrt[n]{a}$	"root a", "the n-th root of a"	The non-negative number whose square (n-th power) is a	$\sqrt[3]{8} = 2$	Chapter 3
$\lvert x \rvert$	"the absolute value of x", "mod x"	Distance from 0	$\lvert -7 \rvert = 7$	Chapter 1
$\lfloor x \rfloor$	"floor of x"	Largest integer $\leq x$	$\lfloor 2.7 \rfloor = 2$, $\lfloor -2.7 \rfloor = -3$	Chapter 11
$\lceil x \rceil$	"ceiling of x"	Smallest integer $\geq x$	$\lceil 2.1 \rceil = 3$	Chapter 11
$n!$	"n factorial"	$1 \cdot 2 \cdots n$, with $0! = 1$	$4! = 24$	Volume 3, Chapter 24

Symbol	Read aloud	Meaning	Example	Ch.
∞	"infinity"	Not a number: shorthand for "without bound"	$x \to \infty$, $(0, \infty)$	Volume 2, Chapter 16
$\ldots$, $\cdots$	"dot dot dot", "and so on"	Continue the evident pattern	$1 + 2 + \cdots + n$	Chapter 14
$\therefore$	"therefore"	Introduces a conclusion	$2x = 6 \therefore x = 3$	Volume 3, Chapter 22
■ or □	"QED", "end of proof"	The proof is complete		Volume 3, Chapter 22

C.2. Number systems and intervals

Symbol	Read aloud	Meaning	Example	Ch.
$\mathbb{N}$	"the naturals", "N"	Natural numbers $\{0, 1, 2, \ldots\}$ (some authors start at 1)	$5 \in \mathbb{N}$	Chapter 1
$\mathbb{Z}$	"the integers", "Z" (from German *Zahlen*)	$\{\ldots, -2, -1, 0, 1, 2, \ldots\}$	$-4 \in \mathbb{Z}$	Chapter 1
$\mathbb{Q}$	"the rationals", "Q" (for *quotient*)	Fractions p/q with p, q integers, $q \neq 0$	$0.75 = \frac{3}{4} \in \mathbb{Q}$	Chapter 1
$\mathbb{R}$	"the reals", "R"	All points on the number line	$\sqrt{2} \in \mathbb{R}$, $\sqrt{2} \notin \mathbb{Q}$	Chapter 1
$\mathbb{C}$	"the complex numbers", "C"	Numbers $a + bi$ with $a, b \in \mathbb{R}$	$3 - 2i \in \mathbb{C}$	Chapter 15
$\mathbb{R}^n$	"R n"	Lists (vectors) of n real numbers	$(1, -2, 0.5) \in \mathbb{R}^3$	Volume 2, Chapter 26
$\mathbb{Z}_m$ or $\mathbb{Z}/m\mathbb{Z}$	"the integers mod m"	The remainders $\{0, 1, \ldots, m-1\}$ with modular arithmetic	In $\mathbb{Z}_{12}$, $9 + 5 = 2$	Volume 3, Chapter 23

Table 2. Interval notation

Notation	Means x satisfies	Endpoints
(a, b)	$a < x < b$	both excluded ("open")
$[a, b]$	$a \le x \le b$	both included ("closed")
$[a, b)$	$a \le x < b$	left included, right excluded ("half-open"; Python's `range`)
$(a, b]$	$a < x \le b$	left excluded, right included
(a, ∞)	$x > a$	∞ is never included
$(-\infty, b]$	$x \le b$	
$(-\infty, \infty)$	any real x	the whole of $\mathbb{R}$

A round bracket means "up to but not including"; a square bracket means "including". Some European texts write $]a, b[$ for (a, b). Do not confuse the open interval (a, b) with the point (a, b) in the plane; context tells them apart. See Chapter 5.

C.3. Sets

Symbol	Read aloud	Meaning	Example	Ch.
$\{\ \}$	"the set containing ..."	Unordered collection, no repeats	$\{1, 2, 3\} = \{3, 1, 2\}$	Volume 3, Chapter 21
$\{x \mid P(x)\}$ or $\{x : P(x)\}$	"the set of x such that P of x"	Set-builder notation: everything satisfying a condition	$\{x \in \mathbb{Z} \mid x^2 < 5\} = \{-2, \ldots, 2\}$	Volume 3, Chapter 21
$\in$	"is in", "is an element of", "belongs to"	Membership	$3 \in \{1, 2, 3\}$	Volume 3, Chapter 21
$\notin$	"is not in"	Non-membership	$4 \notin \{1, 2, 3\}$	Volume 3, Chapter 21
$\subseteq$	"is a subset of"	Every element of the left is in the right	$\{1, 2\} \subseteq \{1, 2, 3\}$	Volume 3, Chapter 21
$\subset$ or $\subsetneq$	"is a proper subset of"	Subset but not equal (some books use $\subset$ for $\subseteq$; check the convention)	$\mathbb{N} \subset \mathbb{Z}$	Volume 3, Chapter 21
$\cup$	"union", "A cup B"	In either set (or both)	$\{1, 2\} \cup \{2, 3\} = \{1, 2, 3\}$	Volume 3, Chapter 21
$\cap$	"intersection", "A cap B"	In both sets	$\{1, 2\} \cap \{2, 3\} = \{2\}$	Volume 3, Chapter 21

Symbol	Read aloud	Meaning	Example	Ch.
$A \setminus B$ or $A - B$	"A minus B", "A without B"	In A but not in B	$\{1,2,3\} \setminus \{2\} = \{1,3\}$	Volume 3, Chapter 21
A^c, $\overline{A}$, A'	"A complement"	Everything (in the universe) not in A	In $\{1,\ldots,6\}$, $\{1,2\}^c = \{3,4,5,6\}$	Volume 3, Chapter 21
$\emptyset$ or $\varnothing$	"the empty set"	The set with no elements	$\{1\} \cap \{2\} = \emptyset$	Volume 3, Chapter 21
$\lvert A \rvert$ or $\#A$	"the size of A", "the cardinality of A"	Number of elements	$\lvert\{a,b,c\}\rvert = 3$	Volume 3, Chapter 21
$A \times B$	"A cross B"	Cartesian product: all ordered pairs (a,b)	$\{1,2\} \times \{x\} = \{(1,x),(2,x)\}$	Volume 3, Chapter 21
$\mathcal{P}(A)$	"the power set of A"	The set of all subsets of A	$\mathcal{P}(\{1,2\}) = \{\emptyset,\{1\},\{2\},\{1,2\}\}$	Volume 3, Chapter 21

C.4. Logic

Symbol	Read aloud	Meaning	Example	Ch.
$\neg p$	"not p"	True exactly when p is false (code: `!p`, `not p`)	$\neg(2 > 3)$ is true	Volume 3, Chapter 21
$p \wedge q$	"p and q"	True when both are true (code: `&&`, `and`)	$x > 0 \wedge x < 1$	Volume 3, Chapter 21
$p \vee q$	"p or q"	True when at least one is true; inclusive or (code: `or`)	$x < 0 \vee x > 1$	Volume 3, Chapter 21
$p \oplus q$	"p xor q", "exclusive or"	True when exactly one is true (code: ^ on booleans)	$\mathbf{T} \oplus \mathbf{T}$ is false	Volume 3, Chapter 21
$p \to q$	"if p then q", "p implies q"	The conditional: false only when p is true and q false	$x > 2 \to x > 1$	Volume 3, Chapter 21
$p \leftrightarrow q$	"p if and only if q", "p iff q"	Same truth value	n even $\leftrightarrow$ n^2 even	Volume 3, Chapter 21
$\Rightarrow$	"implies"	Used between statements in an argument: "from this, it follows that". Many authors use $\Rightarrow$ and $\to$ interchangeably	$2x = 6 \Rightarrow x = 3$	Volume 3, Chapter 21
$\Leftrightarrow$	"if and only if", "is equivalent to"	Each side implies the other; an equation-solving step that can be reversed	$x + 1 = 3 \Leftrightarrow x = 2$	Volume 3, Chapter 21

Symbol	Read aloud	Meaning	Example	Ch.
$\forall$	"for all", "for every"	Universal quantifier	$\forall x \in \mathbb{R},\ x^2 \geq 0$	Volume 3, Chapter 21
$\exists$	"there exists", "for some"	Existential quantifier: at least one	$\exists x \in \mathbb{R},\ x^2 = 2$	Volume 3, Chapter 21
$\exists!$	"there exists a unique"	Exactly one	$\exists! x,\ 2x = 6$	Volume 3, Chapter 21
$\mathbf{T},\ \mathbf{F}$	"true", "false"	Truth values (also $1, 0$ or $\top, \bot$)		Volume 3, Chapter 21

C.5. Functions

Symbol	Read aloud	Meaning	Example	Ch.
$f(x)$	"f of x"	The output of function f at input x (not f times x)	$f(x) = x^2,\ f(3) = 9$	Chapter 11
$f : A \to B$	"f from A to B"	f takes inputs in A (domain) to outputs in B (codomain); a type signature	$\sqrt{\cdot} : [0, \infty) \to \mathbb{R}$	Chapter 11
$x \mapsto f(x)$	"x maps to f of x"	Names a function by its rule (like a lambda)	$x \mapsto 2x + 1$	Chapter 11
$f \circ g$	"f composed with g", "f after g", "f of g"	$(f \circ g)(x) = f(g(x))$: apply g first	$f(x) = x^2,\ g(x) = x + 1 \Rightarrow (f \circ g)(2) = 9$	Chapter 11
f^{-1}	"f inverse"	The function that undoes f. **Not** $1/f$	$f(x) = 2x \Rightarrow f^{-1}(x) = x/2$	Chapter 11
$\sin^{-1} x$, $\arcsin x$	"inverse sine of x", "arc sine of x"	The angle whose sine is x. But $\sin^2 x$ means $(\sin x)^2$	$\arcsin \frac{1}{2} = \frac{\pi}{6}$	Chapter 13
$\log_b x$, $\ln x$, $\log x$	"log base b of x", "natural log of x" ("ell-en x", "lin x" or "log x")	The exponent that gives x. $\ln$ is base e; bare $\log$ is base 10 in school, base e in research papers and in Python, often base 2 in CS	$\log_2 8 = 3$	Chapter 12
$\exp(x)$	"exp of x"	e^x, written so the exponent can be large	$\exp(0) = 1$	Chapter 12
Δx	"delta x"	A change (difference) in x	$\Delta x = x_2 - x_1$	Chapter 5
$F(x)\|_a^b$	"F of x evaluated from a to b"	$F(b) - F(a)$	$x^2\|_1^3 = 9 - 1 = 8$	Volume 2, Chapter 19
$O(g)$, $\Theta(g)$, $\Omega(g)$	"big O of g", "big Theta", "big Omega"	Grows at most / exactly / at least as fast as g, up to a constant factor	Binary search is $O(\log n)$	Chapter 14

C.6. Sequences and sums

Symbol	Read aloud	Meaning	Example	Ch.
a_n	"a sub n", "a n"	The n-th term of a sequence (code: `a[n]`)	$a_n = 2n \Rightarrow a_3 = 6$	Chapter 14
$(a_n)_{n\geq 1}$	"the sequence a n"	The whole sequence, not one term	$(1/n)_{n\geq 1}$	Chapter 14
$\sum_{i=1}^{n} a_i$	"the sum from i equals 1 to n of a sub i"	$a_1 + a_2 + \cdots + a_n$; a `for` loop accumulating with `+`	$\sum_{i=1}^{4} i = 10$	Chapter 14
$\prod_{i=1}^{n} a_i$	"the product from i equals 1 to n of a sub i"	$a_1 \cdot a_2 \cdots a_n$; the same loop accumulating with $\times$	$\prod_{i=1}^{4} i = 24$	Chapter 14
$\binom{n}{k}$	"n choose k"	Number of k-element subsets of an n-element set; also written $C(n,k)$, ${}^nC_k$, ${}_nC_k$, nCr	$\binom{5}{2} = 10$	Volume 3, Chapter 24
$P(n,r)$	"the number of permutations of n things r at a time"	Ordered selections: $n!/(n-r)!$; also ${}^nP_r$, nPr	$P(5,2) = 20$	Volume 3, Chapter 24

C.7. Calculus

Symbol	Read aloud	Meaning	Example	Ch.
$\lim_{x \to a} f(x)$	"the limit as x tends to a of f of x"	The value $f(x)$ approaches as x gets close to (but not equal to) a	$\lim_{x \to 0} \frac{\sin x}{x} = 1$	Volume 2, Chapter 16
$x \to a^+$, $x \to a^-$	"x tends to a from above / below"	One-sided limits	$\lim_{x \to 0^+} \frac{1}{x} = \infty$	Volume 2, Chapter 16
$\varepsilon,\ \delta$	"epsilon", "delta"	Small positive tolerances in the formal definition of a limit		Volume 2, Chapter 16
$f'(x)$	"f prime of x"	The derivative of f at x (Lagrange notation)	$f(x) = x^3 \Rightarrow f'(x) = 3x^2$	Volume 2, Chapter 17
$f''(x)$, $f^{(n)}(x)$	"f double prime of x", "the n-th derivative of f"	Second and n-th derivatives; brackets show it is not a power	$(x^3)'' = 6x$	Volume 2, Chapter 17
$\frac{dy}{dx}$	"d y d x", "the derivative of y with respect to x"	Leibniz notation: rate of change of y per unit x. Not a fraction, though it often behaves like one	$y = x^2 \Rightarrow \frac{dy}{dx} = 2x$	Volume 2, Chapter 17
$\frac{d}{dx}$	"d by d x"	The operation "differentiate with respect to x"	$\frac{d}{dx}\sin x = \cos x$	Volume 2, Chapter 17
$\dot{x}$	"x dot"	Newton notation: derivative with respect to time (physics, mechanics)	$\dot{x} = v$	Volume 2, Chapter 17

Symbol	Read aloud	Meaning	Example	Ch.
$\frac{\partial f}{\partial x}$, f_x, $\partial_x f$	"partial f by partial x", "del f del x", "f sub x"	Partial derivative: differentiate in x, holding the other variables fixed	$f = x^2 y \Rightarrow f_x = 2xy$	Volume 2, Chapter 20
∇f	"grad f", "del f", "nabla f"	The gradient: the vector of all partial derivatives	$\nabla(x^2 + y^2) = (2x, 2y)$	Volume 2, Chapter 20
$\int f(x)\,dx$	"the integral of f of x d x"	Indefinite integral: the family of antiderivatives	$\int 2x\,dx = x^2 + C$	Volume 2, Chapter 19
$\int_a^b f(x)\,dx$	"the integral from a to b of f of x d x"	Definite integral: signed area under f between a and b (a number). The $\int$ is an elongated S, for "sum"	$\int_0^1 x\,dx = \frac{1}{2}$	Volume 2, Chapter 19
dx	"d x"	Marks the variable of integration; the infinitesimal width of a strip		Volume 2, Chapter 19

C.8. Vectors and matrices

Symbol	Read aloud	Meaning	Example	Ch.
$\mathbf{v}$, $\vec{v}$, $\underline{v}$	"vector v", "v"	A vector. Print uses bold; handwriting uses an arrow or underline (the UK schools' squiggle)	$\mathbf{v} = (3, 4)$	Volume 2, Chapter 26
v_i	"v sub i"	The i-th component (code: `v[i]`, but mathematics usually counts from 1)	$v_2 = 4$	Volume 2, Chapter 26
$\|\mathbf{v}\|$	"the norm of v", "the length of v"	Length (Euclidean norm unless a subscript says otherwise)	$\|(3, 4)\| = 5$	Volume 2, Chapter 26
$\hat{\mathbf{v}}$	"v hat"	Unit vector in the direction of $\mathbf{v}$; $\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$ are the axis directions	$\hat{\mathbf{v}} = (0.6, 0.8)$	Volume 2, Chapter 26
$\mathbf{u} \cdot \mathbf{v}$	"u dot v"	Dot (scalar, inner) product; also $\langle \mathbf{u}, \mathbf{v} \rangle$	$(1, 2) \cdot (3, 4) = 11$	Volume 2, Chapter 26
$\mathbf{u} \times \mathbf{v}$	"u cross v"	Cross product (3D only): a vector perpendicular to both	$\hat{\mathbf{i}} \times \hat{\mathbf{j}} = \hat{\mathbf{k}}$	Volume 2, Chapter 26
$\perp$	"is perpendicular to", "is orthogonal to"	At right angles (dot product 0)	$(1, 0) \perp (0, 1)$	Volume 2, Chapter 26
A, a_{ij}	"matrix A", "a i j"	Matrices are capital letters; a_{ij} is the entry in row i, column j		Volume 2, Chapter 27

Symbol	Read aloud	Meaning	Example	Ch.
A^T, $A^\top$, A'	"A transpose"	Rows become columns: $(A^T)_{ij} = a_{ji}$	$\begin{pmatrix}1 & 2\end{pmatrix}^T = \begin{pmatrix}1\\2\end{pmatrix}$	Volume 2, Chapter 27
A^{-1}	"A inverse"	The matrix with $AA^{-1} = I$, when it exists		Volume 2, Chapter 27
I, I_n	"the identity", "I n"	Square matrix with 1s on the diagonal, 0s elsewhere	$I\mathbf{x} = \mathbf{x}$	Volume 2, Chapter 27
$\det A$, $\lvert A\rvert$	"the determinant of A", "det A"	Signed area/volume scale factor of A	$\det \begin{pmatrix}2 & 0\\0 & 3\end{pmatrix} = 6$	Volume 2, Chapter 27
$\operatorname{tr} A$	"the trace of A"	Sum of the diagonal entries	$\operatorname{tr} I_3 = 3$	Volume 2, Chapter 28
λ	"lambda"	An eigenvalue: $A\mathbf{v} = \lambda\mathbf{v}$		Volume 2, Chapter 28

C.9. Probability and statistics

Symbol	Read aloud	Meaning	Example	Ch.
Ω	"omega", "the sample space"	The set of all possible outcomes	Coin: $\Omega = \{H, T\}$	Volume 3, Chapter 29
$P(A)$, $\Pr(A)$	"the probability of A"	A number in $[0, 1]$	$P(\text{even die}) = \frac{1}{2}$	Volume 3, Chapter 29
$P(A \mid B)$	"the probability of A given B"	Probability of A once you know B happened	$P(\text{six} \mid \text{even}) = \frac{1}{3}$	Volume 3, Chapter 29
X, x	"capital X", "little x"	Capital: a random variable; lower case: a particular value it takes	$P(X = x)$	Volume 3, Chapter 29
$X \sim D$	"X is distributed as D", "X follows D"	Names the distribution of X	$X \sim N(0, 1)$	Volume 3, Chapter 30
$E[X]$, $\mathbb{E}[X]$	"the expected value of X", "the expectation of X"	Probability-weighted average	Fair die: $E[X] = 3.5$	Volume 3, Chapter 30
$\text{Var}(X)$	"the variance of X"	$E[(X - \mu)^2]$: average squared distance from the mean	Fair coin (0/1): 0.25	Volume 3, Chapter 30
μ	"mu"	Population (true) mean		Volume 3, Chapter 30

Symbol	Read aloud	Meaning	Example	Ch.
σ, σ^2	"sigma", "sigma squared"	Population standard deviation, variance		Volume 3, Chapter 30
$\text{Cov}(X, Y)$, ρ	"the covariance of X and Y", "rho"	How two variables vary together; ρ is the correlation		Volume 3, Chapter 30
$\bar{x}$	"x bar"	Sample mean	$\overline{\{2, 4, 9\}} = 5$	Volume 3, Chapter 31
s, s^2	"s", "s squared"	Sample standard deviation, sample variance		Volume 3, Chapter 31
$\hat{p}, \hat{y}, \hat{\theta}$	"p hat", "y hat", "theta hat"	An estimate or prediction of the un-hatted quantity	45 of 100 say yes: $\hat{p} = 0.45$	Volume 3, Chapter 31
H_0, H_1	"H nought", "H one"	Null and alternative hypotheses	$H_0 : \mu = 100$	Volume 3, Chapter 31
α	"alpha"	Significance level of a test	$\alpha = 0.05$	Volume 3, Chapter 31
z^*, t^*	"z star", "t star"	Critical values for confidence intervals	$z^* = 1.96$ for 95%	Volume 3, Chapter 31

C.10. Number theory and complex numbers

Symbol	Read aloud	Meaning	Example	Ch.
$a \mid b$	"a divides b"	b is a multiple of a (a statement, not a number)	$3 \mid 12, 5 \nmid 12$	Chapter 1
$a \bmod m$	"a mod m"	The remainder on dividing a by m (code: `a % m`)	$17 \bmod 5 = 2$	Volume 3, Chapter 23
$a \equiv b \pmod m$	"a is congruent to b mod m"	$m \mid (a - b)$: same remainder	$38 \equiv 2 \pmod{12}$	Volume 3, Chapter 23
$\gcd(a, b)$, $\operatorname{lcm}(a, b)$	"g c d of a and b", "l c m"	Greatest common divisor (UK: HCF), least common multiple	$\gcd(12, 18) = 6$	Chapter 1
$\varphi(n)$	"phi of n", "Euler's totient of n"	How many of $1, \ldots, n$ are coprime to n	$\varphi(10) = 4$	Volume 3, Chapter 23
i	"i"	$\sqrt{-1}$; engineers write j, as does Python (`1j`)	$i^2 = -1$	Chapter 15
$\bar{z}$, z^*	"z bar", "the conjugate of z"	Flip the sign of the imaginary part	$\overline{3 + 4i} = 3 - 4i$	Chapter 15
$\lvert z \rvert$, $\arg z$	"mod z", "arg z"	Modulus (length) and argument (angle) of z	$\lvert 3 + 4i \rvert = 5$	Chapter 15
$\operatorname{Re} z$, $\operatorname{Im} z$	"real part of z", "imaginary part of z"	a and b in $z = a + bi$	$\operatorname{Im}(3 + 4i) = 4$	Chapter 15

C.11. Optimisation and machine learning

Symbol	Read aloud	Meaning	Example	Ch.
$\min_x f(x)$, $\max_x f(x)$	"the minimum over x of f of x"	The smallest (largest) *value* of f	$\min_x (x-3)^2 + 1 = 1$	Volume 2, Chapter 18
$\operatorname{argmin}_x f(x)$, $\operatorname{argmax}_x f(x)$	"the arg min over x of f of x"	The *input* x that achieves the minimum (maximum)	$\operatorname{argmin}_x (x-3)^2 + 1 = 3$	Volume 3, Chapter 33
θ^*, x^*	"theta star"	The optimal value of a parameter		Volume 3, Chapter 33
η	"eta"	Learning rate (step size) in gradient descent	$\eta = 0.01$	Volume 2, Chapter 20
$\leftarrow$	"becomes", "gets"	Assignment in pseudocode (code: =)	$x \leftarrow x - \eta f'(x)$	Volume 2, Chapter 32

C.12. The Greek alphabet

Greek letters are used because the Latin ones run out, and certain letters have settled into conventional jobs. Knowing the conventions lets you guess what a symbol means before you're told.

Lower	Upper	Name	Common uses
α	A	alpha	Angles; significance level in statistics; coefficients; sometimes learning rate
β	B	beta	Angles; regression coefficients; Beta distribution; momentum in optimisation
γ	Γ	gamma	Angles; Euler-Mascheroni constant $\gamma \approx 0.5772$; discount factor in reinforcement learning. Γ: gamma function ($\Gamma(n) = (n-1)!$)
δ	Δ	delta	δ: small quantity (ε-δ limits), Kronecker delta. Δ: change in (Δx), discriminant
ε, ϵ	E	epsilon	Small positive quantity; error or noise term; machine epsilon
ζ	Z	zeta	Riemann zeta function $\zeta(s) = \sum 1/n^s$; damping ratio
η	H	eta	Learning rate; efficiency
θ, ϑ	Θ	theta	Angles; model parameters. Θ: asymptotically tight bound in algorithm analysis
ι	I	iota	Rare in mathematics; integer ranges in APL and Go
κ	K	kappa	Curvature; condition number of a matrix; Cohen's kappa
λ	Λ	lambda	Eigenvalues; rate of a Poisson or exponential distribution; Lagrange multiplier; wavelength; anonymous functions (lambda calculus). Λ: diagonal matrix of eigenvalues
μ	M	mu	Mean; micro- (10^{-6}); coefficient of friction
ν	N	nu	Degrees of freedom; frequency
ξ	Ξ	xi	Auxiliary or random variables; the point "somewhere in between" in error formulas
o	O	omicron	Almost never used (indistinguishable from Latin o)
π	Π	pi	$3.14159\ldots$; permutations; policy in reinforcement learning; prime-counting function. Π: product
ρ	P	rho	Correlation; density; radius in polar coordinates

Lower	Upper	Name	Common uses
σ	Σ	sigma	Standard deviation; sigmoid function; singular values; permutations. Σ: summation; covariance matrix; diagonal matrix of singular values in the SVD
τ	T	tau	Time constant; torque; $\tau = 2\pi$; Kendall's rank correlation
υ	Υ	upsilon	Rare
φ, ϕ	Φ	phi	Angles (especially in spherical coordinates); Euler's totient; golden ratio $\varphi \approx 1.618$; standard normal density. Φ: standard normal cumulative distribution function
χ	X	chi	Chi-squared distribution χ^2; characteristic (indicator) function
ψ	Ψ	psi	Wave functions in physics; auxiliary angles and functions
ω	Ω	omega	Angular frequency; complex cube root of unity; an outcome. Ω: sample space; asymptotic lower bound; ohms

Pronunciation: most English speakers say "pie" (π), "fie" or "fee" (φ), "kye" (χ), "zay-ta" or "zee-ta" (ζ), "mew" (μ), "new" (ν), "ksee", "zye" or "sigh" (ξ). US and UK speakers differ on "zee-ta"/"zay-ta" and "bay-ta"/"bee-ta"; either is fine.

C.13. Same symbol, different meanings

Notation is reused shamelessly. When a symbol seems to make no sense, check whether it has switched jobs.

Symbol	Meanings
$\lvert\cdot\rvert$ (vertical bars)	Absolute value $\lvert x\rvert$; modulus of a complex number $\lvert z\rvert$; size of a set $\lvert A\rvert$; determinant $\lvert A\rvert$. A single bar $\mid$ means "divides" ($3 \mid 12$), "such that" ($\{x \mid x > 0\}$) or "given" ($P(A \mid B)$)
Prime $'$	Derivative (f'); a related but different object (A', "A prime"); complement (A'); transpose in statistics (X'); feet and minutes ($5'$)
Bar $\bar{x}$	Mean ($\bar{x}$); complex conjugate ($\bar{z}$); complement ($\overline{A}$); repeating decimal ($0.\overline{3}$); logical NOT in circuit diagrams
Hat $\hat{x}$	Estimate ($\hat{\theta}$); prediction ($\hat{y}$); unit vector ($\hat{\mathbf{v}}$)
Superscript -1	Inverse function (f^{-1}); reciprocal ($x^{-1} = 1/x$); inverse matrix (A^{-1}); modular inverse ($3^{-1} \equiv 5 \pmod 7$)
Superscript 2 on a function	$\sin^2 x = (\sin x)^2$, but $f^2(x)$ sometimes means $f(f(x))$; read the author's conventions
Σ	Summation; covariance matrix; matrix of singular values
π	The constant; a permutation; a policy; the prime-counting function
e	Euler's number $2.71828\ldots$; an event; an edge in a graph; an identity element; scientific notation in code (`1e-9` means 10^{-9})
$\times$	Multiplication; cross product; Cartesian product; dimensions (a 3×4 matrix)
$\equiv$	Identity; congruence; logical equivalence
(a, b)	An open interval; a point in the plane; an ordered pair; $\gcd(a, b)$ in older number-theory texts

C.14. How to read a formula

A dense formula is code with one-letter variable names and no comments. Read it the way you would read unfamiliar code: find the overall shape, identify every name, then trace the evaluation from the inside out.

Example 84. Procedure — decoding a formula

Input: a formula you don't immediately understand.

Steps:

1. **Find the top-level shape.** What is being defined or claimed? Is it "name = expression", an equation to solve, an inequality, an update rule?
2. **Name every symbol.** For each letter, decide: constant, variable, index, function, or operator? Use this appendix and the surrounding text. Which symbols are *bound* (indices of a $\sum$, the variable of an $\int$, the x under an argmin) and which are *free* (inputs the result depends on)?
3. **Read from the inside out**, as an interpreter would evaluate it.
4. **Say it aloud** in words, then in plain English.
5. **Translate it into code.**
6. **Try a tiny example** small enough to do by hand.

Output: a sentence you could say to a colleague, and a few lines of code that compute it.

C.14.1. Decoding 1: sample variance

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Shape. A definition: the name s^2 equals (a fraction) times (a sum).

Symbols. n is the number of data values (free). x_i is the i-th data value. $\bar{x}$ ("x bar") is their mean. i is a bound index: it exists only inside the $\sum$, which runs it from 1 to n. s^2 is a single name, "s squared": it is the square of the standard deviation s.

Inside out. $x_i - \bar{x}$ is how far value i is from the mean (its *deviation*). Squaring makes it non-negative and punishes large deviations more. The $\sum$ adds the squared deviations for every data value. Finally multiply by $\frac{1}{n-1}$: an average, except dividing by $n-1$ rather than n (the reason is in Volume 3, Chapter 31).

Aloud. "s squared equals one over n minus one, times the sum from i equals 1 to n of x sub i minus x bar, all squared."

Plain English. "The sample variance is (almost) the average squared distance of the data from its mean."

Code and tiny example. For the data 2, 4, 6: the mean is 4, the deviations are $-2, 0, 2$, their squares are $4, 0, 4$, the sum is 8, and $8/(3-1) = 4$.

PYTHON

```
xs = [2, 4, 6]
n = len(xs)
xbar = sum(xs) / n
s2 = sum((x - xbar) ** 2 for x in xs) / (n - 1)
print(s2)
```

```
4.0
```

C.14.2. Decoding 2: fitting a model by least squares

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\operatorname{argmin}} \; \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \boldsymbol{\theta}^T \mathbf{x}_i \right)^2$$

Shape. A definition: $\hat{\boldsymbol{\theta}}$ is the result of an argmin, that is, the input that makes some expression smallest.

Symbols. The hat says $\hat{\boldsymbol{\theta}}$ is an *estimate*. Bold says $\boldsymbol{\theta}$ and $\mathbf{x}_i$ are vectors. $\boldsymbol{\theta} \in \mathbb{R}^d$ says $\boldsymbol{\theta}$ is a list of d real numbers (the model's parameters) and that the search ranges over all such lists; $\boldsymbol{\theta}$ is bound by the argmin. $\mathbf{x}_i$ is the i-th input (a feature vector) and y_i the matching observed output. $\boldsymbol{\theta}^T \mathbf{x}_i$ is a transpose times a vector, which is just the dot product $\boldsymbol{\theta} \cdot \mathbf{x}_i$: the model's prediction for example i.

Inside out. $y_i - \boldsymbol{\theta}^T \mathbf{x}_i$ is the *residual*, the truth minus the prediction. Square it, sum over all n examples, divide by n: the mean squared error of the model with parameters $\boldsymbol{\theta}$. The argmin then asks for the parameter vector that makes this error smallest.

Aloud. "Theta hat equals the arg min, over theta in R d, of one over n times the sum from i equals 1 to n of y sub i minus theta transpose x sub i, squared."

Plain English. "Our fitted parameters are whichever parameters make the average squared prediction error on the data as small as possible." This is linear regression (Volume 3, Chapter 31); the argmin is found either exactly by linear algebra (Volume 3, Section 33.2) or by gradient descent (Volume 2, Chapter 20 , Volume 3, Chapter 33).

Code and tiny example. With $d = 1$ and two data points $(x, y) = (1, 2)$ and $(2, 4)$, the loss is $\frac{1}{2}\left((2 - \theta)^2 + (4 - 2\theta)^2\right)$, which is zero at $\theta = 2$, so $\hat{\theta} = 2$. A crude argmin by search confirms it:

PYTHON

```
X = [[1.0], [2.0]]
Y = [2.0, 4.0]

def loss(theta):
    n = len(X)
    return sum((y - sum(t * xj for t, xj in zip(theta, x))) ** 2
               for x, y in zip(X, Y)) / n

candidates = [[k / 100] for k in range(0, 501)]
theta_hat = min(candidates, key=loss)   # argmin by brute force
print(theta_hat, loss(theta_hat))
```

```
[2.0] 0.0
```

Notice that Python's `min(…, key=loss)` returns the *argument*, not the minimum value: it is an argmin. `loss(theta_hat)` is the min.

Appendix D: The US Mathematics Curriculum

If you were educated in the UK, the American system can be baffling: maths is called "math", it is taught as a sequence of separately named year-long courses rather than as one two-year syllabus, and a fifteen-year-old can be "doing Algebra II" or "in AP Calc" depending on which track they are on. This appendix is a translation guide. It explains the structure, compares it with GCSE and A-level, and maps each US course to the chapters of this book that cover it.



Read this as a cross-reference, not a rulebook

The United States has no national curriculum. Each of the 50 states sets its own standards, each of roughly 13,000 school districts chooses its own courses, textbooks and pacing, and individual schools offer different tracks. Everything below describes the *typical* pattern. Course names, grade levels and content vary from place to place; when it matters (for example, when choosing a course for next year), check your school's own course catalogue.

D.1. Grades, ages, and school levels

US school years are called **grades**. A child usually starts **kindergarten** (K) in the year they turn 5 or 6 (the cut-off date varies by state) and finishes **12th grade** at 17 or 18. Because the English Reception year comes before Year 1, US grade n corresponds roughly to English Year $n + 1$.

US grade	Typical age during the year	School level	Rough English equivalent
Pre-K	4-5	Preschool	Reception
K	5-6	Elementary school	Year 1
1-5	6-11	Elementary school	Years 2-6 (KS1-KS2)
6	11-12	Middle school (sometimes elementary)	Year 7
7-8	12-14	Middle school or junior high	Years 8-9 (KS3)
9 ("freshman")	14-15	High school	Year 10
10 ("sophomore")	15-16	High school	Year 11 (GCSE exams)
11 ("junior")	16-17	High school	Year 12 (Lower Sixth)
12 ("senior")	17-18	High school	Year 13 (Upper Sixth; A-level exams)

So a US 15-year-old is usually in 9th or 10th grade, the equivalent of the two GCSE years.

D.2. How US high-school mathematics is organised

The biggest difference from the UK is structural.

Courses, not syllabuses. In England, mathematics from 14 to 16 is one two-year GCSE course ending in national exams, and from 16 to 18 one two-year A-level course ending in national exams. In the US, each school year is a separately named **course** (Algebra I, Geometry, Algebra II, …), each lasting one year, each graded by the teacher, and each recorded on the student's **transcript**.

Continuous assessment. The course grade (A to F) is built from homework, quizzes, unit tests and a final exam set by the school. These grades feed into a **grade point average** (GPA), which universities weigh heavily. There is no national exit exam in mathematics, although many states set **end-of-course** (EOC) tests, especially for Algebra I.

Breadth until 18. US students keep studying six or seven subjects until they graduate. There is no equivalent of choosing three or four A-levels at 16. High schools typically require three or four years of mathematics to graduate (commonly including Algebra I and Geometry, often Algebra II), and selective universities expect four years, ideally reaching Precalculus or Calculus.

Tracks. Students in the same grade can be in very different courses. Placement into an **accelerated** track usually happens around grades 6-7 and determines how far a student gets by 12th grade.

D.3. The typical course sequence

Grade	Standard track	Accelerated track	Highly accelerated
6	Math 6	Math 6 (advanced)	Accelerated Math 6
7	Math 7	Accelerated Math 7 / Pre-Algebra	Pre-Algebra or Algebra I
8	Math 8 / Pre-Algebra	Algebra I	Algebra I or Geometry
9	Algebra I	Geometry (Honors)	Geometry or Algebra II (Honors)
10	Geometry	Algebra II (Honors)	Algebra II or Precalculus (Honors)
11	Algebra II	Precalculus (Honors or AP)	AP Calculus BC
12	Precalculus, Statistics, or another elective	AP Calculus AB or BC; AP Statistics	Multivariable Calculus, Linear Algebra (often via dual enrollment); AP Statistics

The "standard" route is the classic sequence

$$\textbf{Pre-Algebra} \to \textbf{Algebra I} \to \textbf{Geometry} \to \textbf{Algebra II} \to \textbf{Precalculus} \to \textbf{Calculus}$$

and the tracks differ mainly in *when* a student starts it. Taking Algebra I in 8th grade (rather than 9th) is the usual gateway to reaching calculus in high school.

Integrated pathways. A minority of states and districts teach **Integrated Math I, II, III** (sometimes called Math 1-3 or Secondary Math I-III) instead of Algebra I / Geometry / Algebra II. Each year mixes algebra, geometry, functions and statistics, which is closer to the UK approach. The total content over three years is roughly the same.



Why does Geometry get a whole year?

The US "layer-cake" sequence, with a year of Euclidean geometry sandwiched between two years of algebra, dates back to curriculum recommendations made in the 1890s, when Euclid was still the model of rigorous thinking. It survives today, and it has an upside: the US Geometry course is where most American students first write formal proofs, often in a distinctive **two-column** format (statement on the left, reason on the right). A typical UK student meets far less formal proof before A-level.

D.4. Common Core, in brief

The **Common Core State Standards for Mathematics** (2010) are a set of grade-by-grade expectations that most states adopted in the early 2010s. A few states never adopted them (Texas and Virginia, for example, have their own standards), and several have since revised or renamed them, but most state standards still closely resemble Common Core.

Key points for a parent:

- They are **standards** (what a student should be able to do by the end of each grade), not a curriculum or a textbook. Districts choose their own materials.
- Grades K-8 are specified year by year. High school is specified by topic area (Number and Quantity, Algebra, Functions, Modeling, Geometry, Statistics and Probability), so the same standards can be delivered as the traditional sequence or as an integrated one.
- Standards marked (+) are extra topics for students heading towards calculus and STEM subjects, such as complex-number arithmetic, vectors and matrices.
- Eight **Standards for Mathematical Practice** run through every grade: make sense of problems and persevere; reason abstractly and quantitatively; construct viable arguments; model with mathematics; use tools strategically; attend to precision; look for and use structure; look for regularity in repeated reasoning.
- What parents most often notice is the emphasis on multiple strategies and representations (number lines, area models for multiplication, "explain your reasoning"). That emphasis is close in spirit to this book's insistence on connecting words, symbols, pictures and computation.

D.5. Honors, AP, and IB

Honors. An Honors course covers the same course name at greater depth and pace, with harder problems. Many schools give Honors and AP grades extra weight in the GPA.

Advanced Placement (AP). AP courses follow a national syllabus set by the College Board and end in a standardised exam in May, scored 1 to 5. Many universities give credit or advanced placement for a 4 or 5 (some for a 3). AP is the closest US analogue to an A-level: a national, externally assessed exam in one subject. The AP mathematics courses are:

Course	Content	Chapters
AP Precalculus	Polynomial and rational functions; exponential and logarithmic functions; trigonometric and polar functions. (Parametric functions, vectors and matrices appear in the course but not on the exam.)	Chapter 11, Chapter 7, Chapter 12, Chapter 13
AP Calculus AB	Limits and continuity; derivatives and their rules; implicit differentiation; related rates; L'Hôpital's rule; optimisation and curve sketching; Riemann sums; definite and indefinite integrals; the Fundamental Theorem; integration by substitution; slope fields and separable differential equations; area and volume. Roughly a first-semester university calculus course.	Volume 2, Chapter 16 , Volume 2, Chapter 17 , Volume 2, Chapter 18 , Volume 2, Chapter 19
AP Calculus BC	Everything in AB, plus integration by parts and partial fractions; improper integrals; Euler's method; logistic growth; parametric, polar and vector-valued functions; infinite series (convergence tests, Taylor and Maclaurin series). Roughly two semesters of university calculus. The BC exam also reports an AB sub-score.	As AB, plus Chapter 14, Volume 2, Chapter 32
AP Statistics	Exploring and describing data; sampling and experimental design; probability; random variables; binomial and geometric distributions; the normal distribution; sampling distributions; confidence intervals and significance tests (means, proportions, chi-square, regression slope). Roughly an introductory, non-calculus university statistics course.	Volume 3, Chapter 29 , Volume 3, Chapter 30 , Volume 3, Chapter 31 , Volume 3, Chapter 24

Related AP courses that use mathematics heavily: **AP Computer Science A** (Java programming), **AP Computer Science Principles**, **AP Physics 1** (algebra-based mechanics) and **AP Physics C: Mechanics** (calculus-based mechanics).

International Baccalaureate (IB). Some US high schools offer the two-year IB Diploma in grades 11-12. Its mathematics comes in two flavours, **Analysis and Approaches** (more pure, calculus-heavy) and **Applications and Interpretation** (more statistics and modelling), each at Standard Level (SL) or Higher Level (HL). IB Analysis and Approaches HL is roughly comparable to A-level Mathematics plus some Further Mathematics.

Dual enrollment. High-school students can take courses at a local community college or university, earning credit towards both their high-school diploma and a university degree. This is how strong students continue after AP Calculus BC: **Multivariable Calculus**, **Linear Algebra**, **Differential**

Equations and **Discrete Mathematics** are the usual next courses.

D.6. SAT and ACT mathematics

Most US universities consider (and some require) a standardised admissions test, the **SAT** or the **ACT**. Both include mathematics, and both test the content of Algebra I, Geometry and parts of Algebra II, with no calculus. Their formats change from time to time, so check the current specifications, but at the time of writing:

- **SAT Math** (College Board): a computer-adaptive section in two modules, 44 questions in 70 minutes, scored 200-800. A calculator (including a built-in graphing calculator) is allowed throughout. Four content areas: Algebra (linear equations, inequalities and systems); Advanced Math (quadratics, polynomials, exponentials, function notation); Problem-Solving and Data Analysis (ratios, percentages, units, statistics, probability); Geometry and Trigonometry. Relevant chapters: Chapter 2 to Chapter 10, the start of Chapter 13, and the descriptive parts of Volume 3, Chapter 31 .
- **ACT Math**: 45 multiple-choice questions in 50 minutes, scored 1-36, calculator allowed. Slightly broader than the SAT, with occasional logarithms, matrices, complex numbers and trigonometry from Algebra II and Precalculus.
- The **PSAT/NMSQT**, taken in 10th or 11th grade, is a shorter version of the SAT and the qualifying test for National Merit Scholarships.

D.7. The UK and US systems side by side

UK stage	Age	Closest US counterpart	Chapters
Key Stage 3 (Years 7-9)	11-14	Math 6, 7, 8 and Pre-Algebra; Algebra I for accelerated students	Chapter 1 to Chapter 5
GCSE Mathematics, Foundation tier	14-16	Pre-Algebra and Algebra I, with basic geometry and statistics	Chapter 1 to Chapter 6, Chapter 9
GCSE Mathematics, Higher tier	14-16	Algebra I plus Geometry plus parts of Algebra II: quadratics, simultaneous equations, functions, right-triangle trigonometry, sine and cosine rules, circle theorems, basic vectors, probability. No calculus beyond informal gradients and areas under graphs	Chapter 4 to Chapter 10, start of Chapter 13, Volume 3, Chapter 29
Level 2 Further Maths / Additional Maths	15-16	Honors Algebra II and early Precalculus	Chapter 7, Chapter 11, Chapter 13
A-level Mathematics (pure)	16-18	Algebra II, Precalculus and most of AP Calculus BC: proof, functions, logs and exponentials, trigonometric identities, sequences and series, differentiation and integration (including by parts and substitution), parametric equations, simple differential equations, numerical methods (Newton-Raphson, trapezium rule)	Chapter 11 to Volume 2, Chapter 19 , Volume 3, Chapter 22 , Volume 2, Chapter 32
A-level Mathematics (statistics)	16-18	Much of AP Statistics: sampling, binomial and normal distributions, hypothesis testing, correlation and regression	Volume 3, Chapter 29 to Volume 3, Chapter 31
A-level Mathematics (mechanics)	16-18	Taught in *physics* in the US: AP Physics 1, or AP Physics C: Mechanics for the calculus-based version	Volume 2, Chapter 26 , Volume 2, Chapter 17 , Volume 2, Chapter 19

UK stage	Age	Closest US counterpart	Chapters
A-level Further Mathematics	16-18	Beyond the AP courses: complex numbers (including De Moivre), matrices, proof by induction, polar coordinates, hyperbolic functions, series, further calculus and differential equations. Comparable to first-year university Linear Algebra and Differential Equations, reached in the US by dual enrollment	Chapter 15, Volume 2, Chapter 27 , Volume 2, Chapter 28 , Volume 3, Chapter 22 , Chapter 14

Some broad differences are worth keeping in mind.

- **Specialisation.** An A-level student studies mathematics for a large fraction of the school week from 16. A US student spends one period a day on it. In compensation, the US student usually also studies English, history, a science and a language to 18.
- **Calculus timing.** An A-level student meets calculus at 16-17. A US student on the standard track may not meet it until university; an accelerated student meets it at 16-17, much like the A-level student.
- **Proof and geometry.** US students meet formal two-column proof earlier (in Geometry, typically at 14-15). UK students meet more algebraic manipulation earlier.
- **Statistics.** Both systems now include it, but in the US it is often a separate course (AP Statistics) rather than a strand woven through every year.



What happened to "Maths with Mechanics"?

In the 1980s and early 1990s many English A-level students took **Mathematics with Mechanics**, or **Pure and Applied Mathematics**: pure mathematics (algebra, trigonometry, calculus) alongside applied mathematics, which in practice meant Newtonian mechanics (kinematics, forces, Newton's laws, projectiles, moments, energy). The US has no single course like it. The pure half corresponds to Algebra II, Precalculus and AP Calculus; the mechanics half lives in the physics department, where **AP Physics C: Mechanics** is the closest match because it uses calculus.

D.8. Terminology: UK and US words for the same things

UK	US	Notes
maths	math	
Year 11	10th grade	Subtract one from the English year number
BIDMAS / BODMAS	PEMDAS	US mnemonic "Please Excuse My Dear Aunt Sally": Parentheses, Exponents, Multiplication/Division, Addition/Subtraction
brackets	parentheses	US "brackets" usually means square brackets [ ]
highest common factor (HCF)	greatest common factor (GCF) or greatest common divisor (GCD)	
index, indices, powers	exponent(s)	
surd	radical	
standard form (3.2×10^5)	scientific notation	US "standard form" means a line written as $Ax + By = C$, or a number written out in full
$y = mx + c$	$y = mx + b$	The y-intercept is called b in the US
gradient (of a line)	slope	
simultaneous equations	system of equations	
recurring decimal	repeating decimal	
modulus $\lvert x \rvert$	absolute value	"Modulus" is still used for complex numbers
trapezium	trapezoid	Same shape: a quadrilateral with (at least) one pair of parallel sides. Confusingly, in older usage each country's *other* word meant a quadrilateral with no parallel sides
turning point	vertex (of a parabola), extremum	

UK	US	Notes
stationary point	critical point	US "critical point" also includes points where the derivative is undefined
sine rule, cosine rule	law of sines, law of cosines	
Pythagoras' theorem	the Pythagorean theorem	
Newton-Raphson	Newton’s method	
trapezium rule	trapezoidal rule	
sum to infinity	sum of an infinite (geometric) series	
differentiation from first principles	using the limit definition of the derivative	
nought	zero	
marks	points	

D.9. Course-by-course map to this book

Course	Typical grade (standard / accelerated)	Key topics	Chapters
Elementary math	K-5	Place value, the four operations, multiplication tables, fractions (from grade 3), decimals, measurement, simple geometry, simple data displays	Chapter 1, Chapter 2, start of Chapter 9
Math 6 and Math 7	6-7	Ratios, rates and percentages; negative numbers; expressions; one-variable equations and inequalities; area, surface area and volume; introductory statistics and probability	Chapter 1, Chapter 2, Chapter 4, start of Chapter 5, Chapter 9
Math 8 / Pre-Algebra	8 / 7	Exponents and scientific notation; square roots and irrational numbers; linear equations and functions; introductory systems; the Pythagorean theorem; transformations	Chapter 3, Chapter 5, Chapter 6, Chapter 11, Chapter 9
Algebra I	9 / 8	Linear equations and inequalities; slope and lines; functions; systems; exponent rules; exponential functions (introduction); adding, multiplying and factoring polynomials; quadratics; data and scatter plots	Chapter 4, Chapter 5, Chapter 6, start of Chapter 7, Chapter 8, Chapter 11
Geometry	10 / 9	Congruence, similarity and transformations; two-column proof; right-triangle trigonometry; circles; coordinate geometry; area and volume	Chapter 9, Chapter 10, start of Chapter 13, Volume 3, Chapter 22

Course	Typical grade (standard / accelerated)	Key topics	Chapters
Algebra II	11 / 10	Polynomial and rational functions; radicals; complex numbers; exponential and logarithmic functions; sequences and series; inverse functions and transformations; often the unit circle and trigonometric functions; probability and statistics	Chapter 7, Chapter 11, Chapter 12, Chapter 15, Chapter 14, Chapter 13, Volume 3, Chapter 29
Precalculus (including AP Precalculus)	12 / 11	Functions in depth; trigonometric functions, identities and equations; polar coordinates and complex numbers in polar form; vectors; sometimes matrices and conic sections; sequences, series and an introduction to limits	Chapter 11, Chapter 13, Chapter 15, Chapter 10, Chapter 14, Volume 2, Chapter 26 , Volume 2, Chapter 27 , Volume 2, Chapter 16
AP Calculus AB	12 / 11	See the AP table above	Volume 2, Chapter 16 , Volume 2, Chapter 17 , Volume 2, Chapter 18 , Volume 2, Chapter 19
AP Calculus BC	12 / 11	AB plus series, parametric and polar calculus, further integration	As AB, plus Chapter 14, Volume 2, Chapter 32
AP Statistics (or non-AP Statistics)	11-12	Data, sampling and experiments, probability, distributions, inference	Volume 3, Chapter 24 , Volume 3, Chapter 29 , Volume 3, Chapter 30 , Volume 3, Chapter 31
Discrete Mathematics (occasional high-school elective; standard at university)	12 / university	Logic, sets, proof and induction, number theory, counting, graphs	Volume 3, Chapter 21 , Volume 3, Chapter 22 , Volume 3, Chapter 23 , Volume 3, Chapter 24 , Volume 3, Chapter 25

Course	Typical grade (standard / accelerated)	Key topics	Chapters
Multivariable Calculus (Calculus III)	post-BC / university	Partial derivatives, gradients, optimisation in several variables, multiple integrals, vector calculus	Volume 2, Chapter 20 (the book stops before multiple integrals and vector calculus)
Linear Algebra	post-BC / university	Vectors, matrices, systems, determinants, vector spaces, eigenvalues, orthogonality, the SVD	Chapter 6, Volume 2, Chapter 26 , Volume 2, Chapter 27 , Volume 2, Chapter 28

D.10. Quick reference: what does "He's doing ..." mean?

"She's in Math 7"

Ratios, percentages, negative numbers, simple equations. UK Year 8. Chapters Chapter 2, Chapter 4.

"He's doing Pre-Algebra"

The bridge from arithmetic to symbols: exponents, roots, simplifying expressions, solving linear equations, the Pythagorean theorem. UK Years 8-9. Chapters Chapter 3, Chapter 4, Chapter 5.

"She's taking Algebra I in 8th grade"

She is on an accelerated track, one year ahead of the standard sequence, and on course for calculus in high school. Content: lines, systems, functions, factoring, quadratics. Roughly the algebra of GCSE Higher. Chapters Chapter 5 to Chapter 8, Chapter 11.

"He's doing Geometry"

A full year of Euclidean geometry with formal proof, right-triangle trigonometry, circles and coordinate geometry. Roughly the geometry of GCSE Higher, plus far more proof. Chapters Chapter 9, Chapter 10, Volume 3, Chapter 22 .

"He's doing Algebra II"

The big pre-calculus algebra course: polynomials, rational functions, complex numbers, exponentials and logarithms, sequences and series, often trigonometric functions. In UK terms, the top end of GCSE Higher plus a good part of the pure algebra in the first year of A-level. A 15-year-old in Algebra II is a year or two ahead of the standard track. Chapters Chapter 7, Chapter 11, Chapter 12, Chapter 15, Chapter 14.

"She's in Honors Algebra II"

The same territory, faster and with harder problems; sometimes includes topics normally saved for Precalculus.

"He's doing Integrated Math II"

A mixed course: typically quadratics, more functions, similarity and right-triangle trigonometry, probability. Roughly the second half of Algebra I combined with much of Geometry.

"She's doing Precalc"

Functions in depth, full trigonometry, polar form, vectors, an introduction to limits. Roughly A-level pure mathematics without the calculus. Chapters Chapter 11, Chapter 13, Chapter 14, Volume 2, Chapter 16 .

"He's in AP Calc AB" / "She's taking BC"

University-level single-variable calculus, ending in a national exam in May. AB is roughly the calculus of A-level Mathematics; BC adds series and polar and parametric calculus, taking it a little beyond. Chapters Volume 2, Chapter 16 to Volume 2, Chapter 19 .

"She got a 5 in BC"

The top score on the AP Calculus BC exam. Many universities will grant credit for one or two semesters of calculus.

"He's doing AP Stats"

Introductory statistics: data, sampling, probability, the normal distribution, confidence intervals and hypothesis tests. Roughly the statistics of A-level Mathematics. Chapters Volume 3, Chapter 29 to Volume 3, Chapter 31 .

"She's taking Multivariable at the community college"

She has finished AP Calculus BC early and is taking university-level calculus through dual enrollment. Chapter Volume 2, Chapter 20 , and beyond this book.

"He got a 740 on the SAT Math"

A strong score (out of 800), reflecting secure command of Algebra I, Geometry and Algebra II material. It says nothing about calculus.

> ### US curriculum
>
> Every major topic in this book carries a short **US curriculum** sidebar like this one, naming the course where the topic typically appears. Those sidebars and the tables in this appendix are meant to work together: if your teenager says "we're doing logarithms in Algebra II", the sidebar in Chapter 12 tells you where you are, and this appendix tells you what comes next.

D.11. University mathematics for computer science

US computer-science degrees almost all require a common core of mathematics, usually taken in the first two years. (UK computer-science degrees cover much the same ground in first-year modules with names like "Mathematics for Computer Science" and "Discrete Mathematics".)

Course	Typical content	Chapters
Calculus I and II	Limits, derivatives, integrals, techniques of integration, sequences and series (AP Calculus BC with a 4 or 5 often exempts students from both)	Volume 2, Chapter 16 to Volume 2, Chapter 19 , Chapter 14
Discrete Mathematics (one or two semesters)	Logic, sets, functions and relations, proof techniques and induction, recurrences, counting, number theory, graphs and trees. The most important mathematics course for most programmers	Volume 3, Chapter 21 , Volume 3, Chapter 22 , Volume 3, Chapter 23 , Volume 3, Chapter 24 , Volume 3, Chapter 25 , Chapter 14
Linear Algebra	Systems of equations, matrices, vector spaces, determinants, eigenvalues, orthogonality, least squares, the SVD. Essential for graphics, machine learning and scientific computing	Chapter 6, Volume 2, Chapter 26 , Volume 2, Chapter 27 , Volume 2, Chapter 28
Probability and Statistics	Probability, random variables, distributions, expectation, the central limit theorem, estimation and testing; often calculus-based	Volume 3, Chapter 29 , Volume 3, Chapter 30 , Volume 3, Chapter 31
Multivariable Calculus (often optional for CS, required for machine learning, graphics and robotics specialisms)	Partial derivatives, gradients, optimisation, multiple integrals	Volume 2, Chapter 20 , Volume 3, Chapter 33
Numerical Methods / Scientific Computing (elective)	Floating point, root finding, numerical integration and differential equations, numerical linear algebra	Volume 2, Chapter 32

Algorithms and theory-of-computation courses then draw on all of this, especially discrete mathematics, asymptotic notation and probability. Machine-learning courses lean hardest on linear algebra, multivariable calculus, probability and optimisation, which is exactly the combination brought together in Volume 3, Chapter 33 .

Appendix E: Interactive Lab Inventory

Every Interactive Lab opportunity marked in Volume 1, collected in one place. Each entry is a hands-on experiment that makes the idea in its chapter tangible. Most now have a working version on the companion website (https://geekychris.github.io/tools/): follow the entry's _Try it link to open the tool with that example already set up._

There are 65 labs across 15 chapters.

E.1. Chapter 1

Zooming the number line *(in: The number line)*

A number line you can zoom indefinitely. Mark $\sqrt{2}$ and keep zooming: each zoom reveals another decimal digit, and the nearest rational points with small denominators ($\frac{7}{5}$, $\frac{17}{12}$, $\frac{41}{29}$, …) light up and are left behind, but the point itself never lands on a rational.

Try it: Best fractions for √2

Base explorer *(in: Place value and number bases)*

Type a number and watch it displayed simultaneously in bases 2, 8, 10, and 16, with each digit's place value shown as a stack of blocks. Click a bit to toggle it and see the decimal and hex values update; notice that toggling bit k always changes the value by exactly 2^k.

Try it: 255 in every base

Laws as tiles *(in: The four operations and their laws)*

Drag the dividing line in a $a \times (b + c)$ rectangle and watch the two partial areas and their sum update. Then switch to "floating-point mode": generate a random list of 1,000 floats, sum it in forward, reverse, and shuffled order, and display the differing totals.

Try it: Floating-point mode: summation order · Floating-point mode: sum order

Precedence playground *(in: Order of operations)*

Type an expression and see its parse tree drawn, with a selector for convention: "strict PEMDAS", "implied multiplication binds tighter", "Python", "C", "Excel". Watch the tree — and the answer — change for $8 \div 2(2 + 2)$ and -3^2.

Try it: Open the Precedence playground

Factor explorer *(in: Divisibility, primes, and factorisation)*

Enter a number and watch trial division run step by step: the current divisor, the shrinking value, the growing factor list, and a marker showing $\sqrt{n}$ where the search may stop. Then plot the number of steps against n for n up to a million and see primes (worst case) form the top envelope.

Try it: Open the factor explorer · Factorise any number

Overflow clock *(in: Integers inside a computer: two's complement and overflow)*

An n-bit clock with a slider for n from 3 to 8. Each position shows its bit pattern and both its unsigned and signed labels. Add or subtract a chosen amount and watch the hand move; the overflow boundary flashes when crossed. A second mode multiplies, showing that multiplication modulo 2^n also wraps.

Try it: Multiplication wraps too · Open the Overflow clock

Estimation trainer *(in: Estimation and rounding)*

The app shows a quick calculation (387×52, 0.023×3800, $7.9\ \text{GB}/41\ \text{s}$) for three seconds, then asks for an estimate. Score by how close the estimate is on a logarithmic scale, and track your typical error factor over time.

E.2. Chapter 2

Fraction workbench *(in: Arithmetic with fractions)*

Two fraction bars with sliders for numerator and denominator. Press "add" and the app animates re-cutting both bars into the lcm-sized pieces, then slides them together. Press "divide" and it animates tiling the first bar with copies of the second, counting the copies (including a fractional last copy).

Repeating decimal factory *(in: Decimals, and decimals that repeat)*

Enter p and q. The app runs long division with a remainder wheel showing the values 0 to $q - 1$; each step lights up the current remainder, and when one repeats, the cycle is highlighted and the period displayed. Try every denominator from 2 to 50 and look for the pattern in which ones terminate and which have the maximum period $q - 1$ (7, 17, 19, 23, 29, 47, …).

Try it: Open the repeating decimal factory · Repeating decimals of p/q

Percent playground *(in: Percentages)*

Stack a sequence of percentage changes as draggable blocks ($+20\%$, -15%, …) on a starting bar. The bar updates after each block, the running multiplier is shown, and a toggle reveals the naive "add the percents" answer for comparison. Shuffle the blocks and see the final value stay the same.

Float explorer *(in: Fractions inside a computer: why 0.1 is not 0.1)*

Type a decimal number and see its exact stored binary value, its sign/exponent/significand bits, the neighbouring representable doubles on either side, and the error. A second panel sums 0.1 a chosen number of times as float, as `Decimal`, and as integer tenths, plotting the float's drift.

Try it: Watch 0.1 drift · Float explorer: drift of 0.1 · The same drift in float32 · 0.25 is stored exactly

E.3. Chapter 3

Exponent rule checker *(in: The laws of exponents, derived)*

Enter an exponent "rule" — true or false — such as $(a+b)^n = a^n + b^n$ or $a^m a^n = a^{mn}$. The app tests it on random numbers and shows a counterexample if one exists, or a factor-counting picture (rows of coloured tiles for each factor) if it holds.

Root finder ***(in: Roots and fractional exponents)***

Drag a square's side until its area matches a target (say 20) and watch the side length converge; then switch to the Babylonian method and step it, showing each guess and its partner $20/g$ closing in from opposite sides of the true root.

Try it: Babylonian square-root stepper

Scale zoomer ***(in: Scientific notation and significant figures)***

An infinite zoom from 10^{-35} m to 10^{27} m, with landmarks appearing as you pass their order of magnitude. A side panel shows the current scale in scientific notation, E notation, and the matching SI prefix, and lets you pin two objects to compare their ratio.

Bits and bytes calculator ***(in: Powers of two: the computer's native scale)***

A slider for bit width n from 1 to 128 that shows the number of patterns 2^n, the signed and unsigned ranges, the memory addressable with n-bit pointers, and a familiar real-world comparison ("more than the number of grains of sand on Earth"). A second panel converts any size among B, KB, KiB, MB, MiB, …, highlighting the percentage gap.

Try it: Open the Bits and bytes calculator

Growth race ***(in: A first look at exponential growth and logarithms)***

Choose growth functions (n, $n \log n$, n^2, n^3, 2^n, $n!$) and a "computer speed" in steps per second. A race shows the largest n each can handle in one second, one hour, and one year. Toggle the vertical axis between linear and logarithmic and watch the exponential curve become a straight line.

Try it: Growth curves on log axis · Growth race: one second · Growth race: one year

Square-and-multiply stepper ***(in: Exponentiation by squaring)***

Enter a base and exponent. The exponent's bits are shown as a row of switches; stepping through the loop lights up the current bit, shows the square and the result, and counts multiplications against the naive method. A slider pushes the exponent to thousands of bits (with a modulus) to show the naive count becoming astronomically large while the fast count grows by one or two per bit.

Try it: Open the square-and-multiply stepper · Huge powers mod m, instantly

E.4. Chapter 4

Build-a-tree ***(in: Evaluating and substituting)***

Type an expression and see its tree drawn live. Drag values onto the variable leaves and watch each internal node light up with its value as evaluation flows from leaves to root. Then remove a pair of brackets and see how the tree, and the answer, changes.

Resize the rectangle *(in: Expanding brackets: the distributive law)*

Drag sliders for a and b and watch a square of side $a + b$ split into a^2, b^2, and two ab rectangles. Toggle "freshman's dream" to overlay $a^2 + b^2$ alone and see exactly what area is missing.

Balance builder *(in: Equations and the balance model)*

Load an equation onto a virtual balance as boxes (unknowns) and discs (units), including "negative" balloons that pull upward. Apply operations to both pans and watch the beam. Try an illegal move, such as removing a disc from one pan only, and watch it tip. Try "multiply both pans by 0" and watch every possible x become a solution.

Rearrange and verify *(in: Rearranging formulas: changing the subject)*

Pick a formula from physics or finance and a target variable. Apply operations step by step from a palette ("subtract u from both sides", "factor out x"); the lab refuses illegal moves and flags any division by an expression that could be zero. When you finish, it round-trips random values through your rearranged formula to confirm it is correct.

E.5. Chapter 5

Balance-scale solver *(in: Solving linear equations in one unknown)*

Show an equation as a balance scale with x-boxes and unit weights on each pan. The learner chooses an operation ("add 3 to both", "divide both by 2"); the scale applies it to both pans and the symbolic equation updates beside it. Illegal moves, such as acting on one pan only or dividing by zero, visibly tip the scale. A counter compares the learner's number of steps with the minimum.

Inequality explorer *(in: Linear inequalities)*

A slider sets k; the display shows two points $a < b$ on a number line and their images ka and kb. As k passes through 0 the images collide and then swap order, making the sign-flip rule visible. A second panel lets the learner type an inequality and drag a test point, which turns green inside the solution set and red outside.

Slope and intercept sliders *(in: The straight line: $y = mx + b$)*

Sliders for m and b redraw $y = mx + b$ live, with a draggable rise–run triangle whose size changes while its ratio stays fixed. A second mode shows two draggable points; the line through them and its three equation forms update as the points move. A third mode shades the region where $mx + b > k$ on the x-axis.

Try it: Slide m and b

Lerp playground *(in: Linear interpolation)*

Drag two points and a slider for t from -0.5 to 1.5; the interpolated point moves along the line, turning red when it leaves $[0, 1]$ (extrapolation). A second panel adds a third control point and shows the nested lerps of de Casteljau's construction tracing a Bézier curve.

E.6. Chapter 6

Drag the lines *(in: Pictures first: the graphical method)*

Show two lines with sliders for each slope and intercept, and display the intersection point and its coordinates live. Rotate one line until it's parallel to the other and watch the intersection run off to infinity and vanish. Then slide its intercept until the lines coincide. A second mode shows the equations in standard form $ax + by = c$ with sliders on a, b, c, so the learner sees how coefficients control slope.

Try it: Two lines with sliders

Watch the staircase form *(in: Gaussian elimination: the algorithm)*

Display a 3×3 or 4×4 augmented matrix with a "step" button. Each press performs one row operation and highlights the pivot, the row being changed, and the multiplier. Alongside, a 3D view shows the three planes; after each row operation the planes **move** but their common point stays fixed. That's the geometric meaning of "equivalent system". Include a toggle between hand-friendly pivoting (smallest nonzero integer) and partial pivoting (largest absolute value), and a mode where the learner chooses each operation and the app checks it.

Slide the profit line *(in: Systems of linear inequalities and feasible regions)*

The learner enters up to five linear inequalities. The app shades the feasible region and labels its vertices. A draggable slider for P moves the objective level line $30x + 20y = P$ across the plane, and the line turns red the moment it leaves the region. A second control rotates the objective (changing the profit ratio) so the learner can watch the optimal vertex jump from corner to corner, and see the tie when the objective lines up with an edge.

Try it: Slide the profit line

E.7. Chapter 7

Grid multiplier *(in: Adding, subtracting and multiplying polynomials)*

Type two polynomials; the lab draws the area-model grid with each cell sized by its term (for positive values of x) and colours cells of equal degree the same. Slide x and watch the total area equal the product of the side lengths.

Factor explorer *(in: Factoring)*

Drag the two binomials $(px + q)$ and $(rx + s)$ by adjusting p, q, r, s; the lab shows the expanded trinomial and its area model. Then flip modes: given a trinomial, rearrange the tiles of an algebra-tile set into a rectangle, or discover that no rectangle exists.

Horner versus the naive loop *(in: Evaluating polynomials fast: Horner's method)*

Pick a degree n and a value of x. The lab evaluates the polynomial three ways, counts multiplications, and (in floating point, for polynomials like $(x-1)^{10}$ expanded out) plots the error of each method near $x = 1$, where cancellation makes the expanded form noisy whichever way you evaluate it: the naive method is worst and Horner is somewhat better, while evaluating the factored form $(x-1)^{10}$ directly stays accurate.

Try it: Horner versus the naive loop

Hash collisions by design ***(in: Computer-science connections: hashing and checksums)***

Choose a small modulus M (say 101) and a base B. Type strings and watch their polynomial hashes; the lab searches for two strings with the same hash and shows the difference polynomial having B as a root mod M. Then increase M and watch collisions become rare.

E.8. Chapter 8

Four views at once ***(in: One object, four representations)***

Three sliders for a, b, c. Four synchronised panels: the equation in standard, vertex and factored form (factored form greys out when there are no real roots); the parabola with vertex, axis and roots marked; a table of values with first and second differences; and a short sentence in words ("opens downward, maximum 13.48 at t = 1.53, two real roots"). Drag the graph's vertex directly and watch the coefficients change.

Try it: Open the Four views lab · Parabola with a, b, c sliders

Discriminant explorer ***(in: The discriminant: how many roots, before you find them)***

Fix a and b; slide c. Watch the parabola rise, the two roots slide together, merge at $\Delta = 0$, and vanish. A second panel plots the roots in the **complex plane**: as they leave the real axis they become a symmetric pair $p \pm qi$ moving vertically, preparing the ground for Chapter 15.

Try it: Open the Discriminant explorer · Open the discriminant explorer · Slide c, watch roots merge

Bisection versus Newton ***(in: Method 4 and 5: graphing and numerical root finding)***

Plot a quadratic and run bisection and Newton side by side, one step per click. Bisection shows a shrinking bracket; Newton shows tangent lines. Choose x_0 exactly at the vertex and watch Newton fail (horizontal tangent, division by zero); start Newton on $x^2 + 1$ and watch it wander chaotically, because there is no real root to find.

Try it: Bisection versus Newton, step by step · Newton on $x^2 + 1$ · Newton from the vertex

How much precision do you lose? ***(in: When the formula fails: catastrophic cancellation)***

Slide b from 10 to 10^{10} in $x^2 + bx + 1 = 0$. Show both formulas' results digit by digit, with correct digits in green and wrong digits in red, and the number of lost digits plotted against $\log_{10} |b|$. Add a "precision" switch (16, 7, or 4 significant digits) to see the same collapse happen earlier in single precision and in the by-hand 4-digit model.

Try it: How much precision do you lose?

A one-sphere ray tracer *(in: Computer-science connection: ray–sphere intersection)*

A 200 by 200 pixel canvas. For each pixel, fire a ray and colour it by the discriminant: black for $\Delta < 0$, and shaded by t_1 (near is bright) otherwise. Move the camera into the sphere and watch the $t_1 < 0 < t_2$ case take over. Then toggle "textbook formula" and push the sphere far away to see cancellation noise appear at the edges.

E.9. Chapter 9

Shadow measurer *(in: Similarity and scale factors)*

Drag the sun's elevation angle and the height of a pole. Watch the shadow length change, and see the "height / shadow" ratio stay constant for every object in the scene at a given sun angle. Then drag a second slider that scales a triangle by k and read off its perimeter ($\times k$) and area ($\times k^2$) live.

Pythagoras jigsaw *(in: The Pythagorean theorem)*

Drag the corner of a right triangle to change a and b. The app animates the four triangles sliding between the two arrangements, and displays a^2, b^2 and c^2 as numbers that always balance. Then unlock the right angle: as the angle opens past 90°, watch c^2 exceed $a^2 + b^2$.

Try it: Solve any triangle

Inscribed angle explorer *(in: Circles)*

Fix points A and B on a circle and drag a third point V around the rim. The display shows $\angle AVB$ and the central angle $\angle AOB$ live. Watch $\angle AVB$ stay constant while V stays on one arc and jump to $180° - \angle AVB$ when it crosses to the other. Drag A and B to opposite ends of a diameter to see Thales' right angle.

Try it: Open the inscribed angle explorer

Cube into three pyramids *(in: Surface area and volume)*

A 3D model of a cube that splits along its interior diagonals into three congruent square pyramids, which can be dragged apart and rotated, then reassembled. A second panel fills a cone and a cylinder of the same base and height with "water" to show three cones fill one cylinder.

Try it: Cylinder: three cones' volume

E.10. Chapter 10

Drag the triangle *(in: Distance and midpoint)*

Drag two points around the plane. The app draws the right triangle with legs Δx and Δy, shows $\Delta x^2 + \Delta y^2 = d^2$ updating live, and marks the point $P(t)$ as you move a slider for t from 0 to 1. Try dragging one point across the other and watch Δx change sign while d stays positive.

Focus and directrix playground *(in: Conic sections)*

Place a focus and a directrix, then drag a slider for eccentricity e. The app plots every point whose focus-distance is e times its directrix-distance. Watch the ellipse stretch as $e \to 1$, snap into a parabola at $e = 1$, and split into a hyperbola beyond. A second mode slices a 3D cone with a tiltable plane.

Try it: Open the focus–directrix playground · Slide the eccentricity

Transformation sandbox ***(in: Transformations as coordinate rules)***

Draw a triangle, then stack transformations (translate, reflect, rotate, dilate) in a list you can reorder. The app shows each intermediate image and the final rule $(x, y) \mapsto (x', y')$. Swap two items and see whether the result changes. A matrix panel shows the 2×2 (or 3×3 homogeneous) matrix of the composite.

Try it: Transformation sandbox

Pixel-by-pixel ***(in: Drawing a line on a pixel grid: Bresenham's algorithm)***

Click two pixels on a zoomed-in grid. Step through Bresenham one column at a time, watching the midpoint dot, the true line, and the value of D in a side panel. Toggle to DDA and to anti-aliased mode and compare the results on near-horizontal, 45°, and steep lines.

Try it: Open the pixel-by-pixel lab

Unit circles in three metrics ***(in: Bounding boxes, point-in-circle tests, and distance metrics)***

Pick a metric (Euclidean, Manhattan, Chebyshev, or a general L^p with a slider for p) and the app draws every point at distance 1 from a centre you can drag. Watch the circle morph from diamond ($p = 1$) through round ($p = 2$) to square ($p \to \infty$). A second panel runs A* on a grid with each metric as heuristic and counts the nodes explored.

Try it: When p < 1 breaks · Morph the unit circle, run A* · Morph the L^p unit circle · Morph the unit circle with p

E.11. Chapter 11

Draw-and-test ***(in: Graphs and the vertical line test)***

Sketch any curve freehand on a coordinate grid. A vertical line follows the cursor and lights red whenever it meets the curve more than once; the app then offers to split the curve into the fewest function-pieces. Try a circle, a sideways parabola $x = y^2$, and the letter S.

Family portraits ***(in: Families of functions)***

Choose a family and drag sliders for its parameters (m, b for lines, a, b, c for quadratics, k, n for powers). A second panel shows the table of values and its first and second differences, so you can see "constant first differences = linear, constant second differences = quadratic" happening live.

Try it: Quadratic family with table

Four sliders *(in: Transformations of graphs)*

Pick a parent (x^2, $|x|$, $\sqrt{x}$, x^3, $\lfloor x \rfloor$) and drag sliders for a, b, h, k in $af(b(x-h))+k$. Ghost arrows show where each key point of the parent goes. Then try "guess the formula": the app shows a transformed graph and you set the sliders to match it.

Try it: a f(b(x − h)) + k sliders

Mirror in $y = x$ *(in: Inverse functions)*

Draw a function's graph; the app draws its reflection in $y = x$ and colours it red wherever the reflection fails the vertical line test. Drag a "domain window" to restrict the original until the reflection becomes a valid function.

Try it: Reflect y = x² in y = x

Fit a family *(in: Functions as models)*

Load a small data set (stopping distance against speed, average cost against batch size, height against time). Choose a family, drag its parameters to fit the points, and watch a residual plot. Then drag a slider extending the x-range beyond the data and see which fitted families stay sensible and which go wild.

Try it: Fit braking distances

Collision finder *(in: Functions in computing)*

Type keys into a hash table with an adjustable number of buckets and a choice of hash function (length of string, sum of character codes, CRC32). Watch the buckets fill, count collisions, and compare against the "birthday" prediction that collisions appear after roughly $\sqrt{n}$ keys in n buckets (Volume 3, Chapter 29). Then switch the codomain to as many buckets as keys and try to find a hash that is a bijection.

E.12. Chapter 12

Two-point exponential fit *(in: Exponential models and fitting them to data)*

Drag points on a scatter plot and switch the y-axis between linear and log scale. Watch exponential data straighten under the log. Fit a line on the log view and see the fitted a and k (and doubling time) update live. Then add a logistic "saturation" and watch the fit's predictions go badly wrong beyond the data.

Try it: Exponential fit to data

Halving explorer *(in: Logarithms in computer science)*

Pick n with a slider (1 to 10^{12}) and watch binary search on a bar of n items. The bar halves on each step and a counter shows comparisons. Beside it, show $\log_2 n$, the bits needed for n, and the height of a balanced tree holding n keys, so the reader sees all three are the same number.

Try it: Halving explorer

Growth-race plotter ***(in: Logarithms in computer science)***

Plot $\log n$, n, $n \log n$, n^k (with k on a slider) and b^n (with b on a slider). Toggle linear, semi-log and log-log axes. Find where 1.01^n finally overtakes n^3. (It happens near $n \approx 2{,}340$.)

Try it: Growth race, log-log axes · Growth-race plotter · 1.01ⁿ against n³

E.13. Chapter 13

Spin the unit circle ***(in: The unit circle: trigonometry for every angle)***

Drag a point around the unit circle. Show its coordinates $(\cos\theta, \sin\theta)$ live, colour the quadrant, and draw the reference angle. Next to it, trace $\sin\theta$ and $\cos\theta$ as graphs against θ as the point moves, so the reader sees the circle "unwrap" into waves. Add a toggle between degrees and radians.

Try it: Unit circle unwrapped · Spin the unit circle

Polar sketcher ***(in: Polar coordinates)***

Type $r = f(\theta)$ (for example $\cos(k\theta)$ with k on a slider) and watch the curve trace out as θ sweeps from 0 to 2π, with the radius line drawn live. Integer k gives roses with k or $2k$ petals. Find the rule.

Try it: Rose curves r = cos(kθ)

Build a synth ***(in: Trigonometry in computing)***

Sliders for the amplitudes of the first eight harmonics of a 220 Hz tone. Show the summed waveform and **play** it live. Presets: odd harmonics with amplitude $1/k$ (approaching a square wave), all harmonics with $1/k$ (a sawtooth). Then show the FFT bar chart of the result so synthesis and analysis appear side by side.

Try it: Harmonics towards a square wave · All harmonics make a sawtooth · Odd harmonics make a square

Rotate and aim ***(in: Trigonometry in computing)***

A turret at the centre of the screen tracks the mouse. Toggle between `atan(dy/dx)` and `atan2(dy, dx)` and watch the naive version flip the wrong way in two quadrants. Add a toggle for "y points down" to see the clockwise/anticlockwise flip, and a "degrees bug" toggle that passes degrees to `sin` / `cos`.

Try it: Rotate and aim

E.14. Chapter 14

Geometric series slider ***(in: Infinite geometric series)***

Choose a and drag a slider for r from -1.5 to 1.5. Plot the partial sums $S_1, \ldots, S_{40}$ as dots, with a horizontal line at $\frac{a}{1-r}$ when $|r| < 1$. Watch the dots zig-zag in when $r < 0$, creep in slowly when r is close to 1, and break away to infinity the moment $|r|$ reaches 1.

Try it: Partial sums, slide the ratio

Visualise the waste ***(in: Sums and recurrences in algorithm analysis)***

Animate the call tree of `fib(n)` for n up to 10, colouring nodes by argument. Toggle "memoise" and watch the tree collapse into a single path with short side-branches. Display call counts side by side with $2F_{n+1} - 1$ and with $n + 1$.

Try it: Memoised fib collapses · Visualise the waste · Count the calls in Python

Gauss's staircase, live ***(in: Sums and recurrences in algorithm analysis)***

Drag a slider for n and watch two staircases of 1 to n blocks slide together into an $n \times (n + 1)$ rectangle. A second mode shows the nested-loop grid filling cell by cell as a program runs, with a running counter matching $\frac{n(n-1)}{2}$.

Try it: Staircase against n(n + 1)/2 · Count the loop in Python

E.15. Chapter 15

Euler's formula explorer ***(in: Euler's formula)***

Drag θ from 0 to 2π. Show the point $e^{i\theta}$ travelling around the unit circle, with its shadow on the real axis tracing $\cos\theta$ and its shadow on the imaginary axis tracing $\sin\theta$ as two graphs unrolling beside the circle. A second panel shows the partial sums of the power series $\sum (i\theta)^n/n!$ as a chain of arrows, each turned $90°$ from the last and shrinking, spiralling into the point $e^{i\theta}$.

Try it: Open the Euler formula explorer · e^(iθ) and its shadows

Explore the Mandelbrot set ***(in: The Mandelbrot set)***

Click any point c in the plane and watch the orbit $0, c, c^2 + c, \ldots$ drawn as a path of connected dots: spiralling into a fixed point, cycling, or flying off. Zoom into the boundary with a scroll wheel and a slider for the iteration limit, and see where double precision gives out.

Try it: Explore the Mandelbrot set

Multiply and watch ***(in: The Mandelbrot set)***

Draw a shape (a letter F works well, since it has no symmetry). Drag a complex number w around the plane and watch the shape multiplied by w update live: rotating as w moves around a circle, growing and shrinking as w moves in and out, and turning exactly a quarter-turn when $w = i$. A second mode shows $z \mapsto z^2$ distorting a grid, doubling all angles.

Try it: Multiply and watch

Version 0.1
Last updated 2026-10-14 13:05:05 -0700