

Writing mathematics in sentences

Last modified: 2026-10-28 01:12:24 (PDT)

AI-Assisted Authorship

This chapter was written with the assistance of GitHub Copilot, which expanded on outlined ideas and draft notes provided by the author. The content represents the author's perspective and has been reviewed for accuracy, but the detailed prose was generated through AI assistance.

A mathematical paper is made of sentences, and the formulas are part of the sentences. This chapter collects rules for fitting formulas into prose. It draws on the lecture notes of a Stanford course on mathematical writing (Knuth et al. 1987). This chapter selects the rules that bear most on formulas. Several of the course's other rules appear in other chapters, and Table 1 points to them. The notes also cover spelling, punctuation and refereeing, which this book does not.

Keep formulas inside sentences

Readers encounter a formula as part of a sentence, so the words around it decide whether the formula can be read.

Separate formulas with words

Put words between two formulas. Without them, the reader cannot tell where one formula ends and the next begins.

Example 0.1 (Separating two formulas).

✗ Define f_k , $k \geq 2$.

✓ Define f_k , where $k \geq 2$.

In the ✗ version the comma between the two formulas reads as part of the first one. The ✓ version names the relation between them.

Do not start a sentence with a symbol

A sentence that starts with a symbol is hard to see as a new sentence, because the formula runs on from the period before it. Start with a word, such as the name of the object.

Example 0.2 (Starting a sentence with a word).

✗ $t^2 - 5$ has two real roots.

✓ The polynomial $t^2 - 5$ has two real roots.

Write logical symbols as words

In prose, write “implies”, “for every” and “there exists”, not $\Rightarrow$, $\forall$ and $\exists$. The same goes for the symbols for “therefore” and “such that”. Keep the symbols in a work on logic.

Test the sentence with the formulas removed

Many readers skim past the formulas on a first reading. Replace each formula by a placeholder such as “blah”. If the sentence no longer reads smoothly, rewrite it.

Tie formulas together with commentary

A list of formulas with no words between them is the style of a homework paper. Say what each step does and why the next one follows. A sentence such as “Because the set S has an element, we can pick one and call it s ” tells the reader where the argument is going, and the bare formula does not.

Make the prose say what the formula says

The sentence that introduces or interprets a formula must describe that formula. Check every term in the sentence against the formula, symbol by symbol. The words and the symbols must name the same quantity, the same data, and the same kind of average. When you change one, change the other in the same edit.

Example 0.3 (A sentence that names the wrong average). Suppose the formula is $\mathbb{E}[(Y_* - \hat{y}(x))^2]$, a [population](#) mean at one covariate value x . The sentence “the average squared error over the training set” does not describe it. It names a sample average over many observations. Write “the expected squared prediction error at x ” instead.

Before you submit, reread each formula next to the sentence that describes it. Do the same after every edit to either one. Related rules cover [using one notation for one thing](#) and [defining every variable when you introduce it](#).

Introduce a theorem with a sentence

The line before a theorem, lemma or algorithm should be a complete sentence, or end with a colon. Do not let a heading such as “Theorem” finish a sentence that began above it.

Example 0.4 (Introducing a theorem).

✗ This gives the following **Theorem**. $G(x)$ is increasing.

✓ Next we prove a result about monotonicity. **Theorem**. The function $G(x)$ defined in equation (3) is increasing.

The ✗ version breaks a sentence across the theorem and starts the theorem with a symbol. The ✓ version uses a complete sentence, and the theorem names its object.

Write the statement of a theorem so that it stands alone. A reader who jumps to the theorem should not need the assumptions in the paragraph before it, so repeat them in the statement.

Display and number the formulas that matter

Put each important formula on a line by itself. Number the important ones, even if you do not refer to them, because a reader may want to refer to them later. Quarto’s [equation cross-references](#) number a display and let you link to it.

Write numbers according to how they are used

Spell out a small number when it is an adjective: “the proof uses three lemmas”. Write the numeral when the number is a name or a value: “Lemma 3 is used 12 times”, “the total rose by 4”. Capitalize the name of a numbered result, as in “Theorem 1” and “Lemma 2”.

Keep subscripts under control

Do not name the elements of a set unless you need to. If you write “Let $X = \{x_1, \dots, x_n\}$ ”, you will need $x_{i_1}, \dots, x_{i_m}$ as soon as you take a subset of X . A subscript on a subscript is hard to read. Write “an element x of X ” instead, and refer to x and y for two elements. Use x_1 and x_2 only when you need two specified elements.

Choose the voice and the person

Use “we” to mean the author and the reader together, as in “we can now prove”. Write “we prove”, not the passive “it can be proved”. Avoid “I” unless the author’s persona is relevant.

Write “assume that G is finite”, not “assume G is finite”. The word “that” helps the reader parse the sentence. Do not write “we have that $y = z$ ”; write “we have $y = z$ ”.

Choose the proof that fits the claim

Do not use proof by contradiction where a direct proof will do. The course’s worked example proves a lemma both ways. The direct proof is shorter, and it does not assume that the entries are integers or restate their ordering.

Cite results, not solutions

A proof, or any derivation that supports a result, never cites a solution to an exercise. A solution answers an exercise for the reader who attempted the exercise, so the solution is not a result that other arguments can build on. Follow each solution that contains a reusable result with a result div (a theorem, lemma, corollary or proposition) that states the claim. The test is whether the solution contains a result that other arguments could use, not whether any later argument currently relies on it. Cite that result div in every argument that uses the claim.

Example 0.5 (Citing a result div).

✗ **Proof.** By Solution 9 and Solution 10, the claim holds.

✓ **Proof.** By Lemma 4 and Lemma 5, the claim holds.

The ✗ version sends the reader to answers that the notes may reorder, hide or drop. The ✓ version cites claims that the notes state, number and keep.

Add an exercise and its solution just before every result div that you add or edit. The exercise does not state the result. It asks the student to find the result, such as “find an expression for the spread of a random variable using its first two moments”, so that the reader discovers the

result, and the result div then states what the solution showed. Every later argument cites the result div.

The one exception to the citation rule follows from that pairing: the proof of the result itself may cite the solution of the exercise that asks for that same result. Such a proof does no derivation of its own: it is the one line `By @sol-x.`, with one citation. This is the one place where a proof cites a solution. [Derivation format](#) describes the exercise, solution, theorem and proof pattern. When the exercise has several parts, the line may cite each part, as in the line `By @sol-x and @sol-x-part2.`, which ends in a period. A proof with its own steps or display math cites result divs instead.

The `check-solution-citations` workflow in [Morrison-Lab/gha](#) flags a `@sol-` cross-reference inside a proof or result div. It lets a one-line proof with one citation pass. For a one-line proof that cites several parts of the paired solution, the HTML comment `<!-- check-solution-citations: allow -->` on that line tells the workflow to skip that line. Quarto removes the comment from the rendered page. The workflow does not check prose outside proof and result divs, but the rule still applies there.

Keep each div minimal: cite, never inline

Every result div states one result and every solution div answers one exercise, and each does no more. When a proof or a solution needs a step that a reader could use elsewhere, state that step as its own result div, preceded by its exercise and solution, and cite that result div. Never derive the step inline. A solution decomposes the same way: each solution is the smallest argument that answers its exercise, and it cites result divs for the steps that other arguments also use.

Example 0.6 (Citing a step instead of deriving it).

✗ Proof. By the chain rule, the gradient of each summand is $2\hat{\varepsilon}_i D(\tilde{x}_i)$, where $\hat{\varepsilon}_i$ is the residual and $D(\tilde{x}_i)$ the design vector, and summing over i gives the claim.

✓ Proof. Lemma 4 gives the gradient of each summand, and Lemma 5 says that the gradient of a sum is the sum of the gradients, so the gradient of the total is the sum of $2\hat{\varepsilon}_i D(\tilde{x}_i)$ over i .

The **✗** proof derives inline the result that Lemma 4 states, so a reader who needs that step elsewhere has nowhere to cite.

No div repeats reasoning that it could cite instead. Two arguments that share a step cite one result div, so a correction to the step is made once.

No check can decide whether a step is reusable, so this rule stays a writing rule that review enforces. The `check-solution-citations` workflow covers only the citation of solutions.

Rules that other chapters cover

Table 1 lists the course’s other rules and where this book discusses them.

Table 1: Rules from the course that other chapters of this book cover

Rule	Where it appears
Define every variable when you introduce it	Defining terms
Use one notation for one thing, and one thing for each notation	Mathematical notation
Motivate a result before you derive it	Show why you need a result before you derive it
Avoid strings of nouns used as adjectives	Rewrite noun strings and hidden verbs
Avoid unnecessary jargon	Minimize unnecessary jargon
Write sentences that can be read from left to right without ambiguity	Avoid garden-path sentences
Use “that” instead of “which” in a restrictive clause	Relative pronouns
Do not praise your own work	Tone

Knuth, Donald E., Tracy Larrabee, and Paul M. Roberts. 1987. *Mathematical Writing*. https://jmlr.csail.mit.edu/reviewing-papers/knuth_mathematical_writing.pdf.