

L^AT_EX Style Guide for EE 364B

Stephen Boyd Ernest K. Ryu Neal Parikh

May 12, 2014

Abstract

This document lays out rules and guidelines for mathematical writing and typesetting that you must follow for your project proposals, reports, and the final exam, all of which *must* be typeset in L^AT_EX. The guidelines here cover both the L^AT_EX source as well as the output, so this PDF is intended to be read alongside its own source code.

There are many well-known references on mathematical writing, but the goal here is to briefly survey this topic and to lay out some rules specific to this course. Parts of this document were inspired by or lifted from Halmos [Hal70] and Knuth [KLR89].

Contents

1	Style guidelines	3
2	Source code	10
3	Content	11

While you are writing, it is often useful to include the table of contents so you can see the structure of your document, even if you don't include it in the final version.

1 Style guidelines

You will likely find that attempting to write things out will reveal that you were not as clear about your topic as you thought you were. John von Neumann once said, "There's no sense in being precise when you don't even know what you're talking about," and Niels Bohr wrote, "Never express yourself more clearly than you can think." Keep these in mind.

Many respectable books follow similar rules, like [BV04], [CT91, p. 23], [HTF01, p. 26], [Sip01, p. 21], [CSRL01, p. 25], [Rud76, p. 15], [Eva10, p. 18], [Gol80, p. 3], and [Knu73].

Write good English. Always write good English, "even" when the subject is mathematics. This includes correct grammar, word choice, punctuation, spelling, phrasing, and common sense. A classic on this topic, only slightly dated, is Strunk and White [SW57].

Keep the reader in mind. Perhaps the most important principle of good writing is to keep the reader in mind: What do they know so far? What do they expect next and why? Do they have sufficient motivation for stated results? As part of this, make sure you know what level of reader you are writing for and stay consistent with that level. If the reader is expected to know convex analysis, do not keep defining standard concepts like subgradients.

Write to allow skipping over formulas. Many readers will first read through the paper ignoring or skipping all but the simplest formulas. Your sentences and overall paper should flow smoothly, and make sense, when all but the simplest formulas are replaced by "blah" or a similar placeholder. As a related point, do not simply display a list of formulas or equations in a row; tie the concepts together with a running commentary.

Be precise with your language. The sentence "Let x^* be the solution to the optimization problem" implicitly asserts that the solution is unique. If the solution is not unique or need not be unique, write, "Let x^* be a solution to the optimization problem." Similarly, do not refer to "solving" an expression, as this is meaningless. You can solve an equation or set of equations, evaluate an expression or function, or check that an equation or inequality holds.

Use the editorial we when appropriate. The word "we" is often useful to avoid passive voice, which can sometimes be awkward, or the use of "I" or "one", which should be avoided. However, be careful not to overuse "we", as it can become a very bad habit.

Often bad: We can see that Theorem 1 implies Corollary 2.

Good: Theorem 1 implies Corollary 2.

In general, use whatever phrasing makes the sentence cleaner and less clunky. To quote Strunk and White, "Omit needless words."

Punctuation in equations. An equation is part of a sentence, and you may need to include a comma or a period at the end of an equation as a result, whether or not you are using inline or display math style. For example:

We next discuss how to solve the problem

$$\text{minimize } (1/2)\|Ax - b\|_2^2,$$

where $x \in \mathbf{R}^n$ is the optimization variable.

Don't start a sentence with a symbol. This hurts readability:

Bad: f is smooth.

Good: The function f is smooth.

Bad: $x^n - a$ has n distinct zeros.

Good: The polynomial $x^n - a$ has n distinct zeros.

Similarly, don't start a sentence with a reference.

Use words to separate symbols in different formulas. If it might confuse the reader visually or in the actual meaning of the sentence, use words to break apart formulas:

Bad: The sequences $x_1, x_2, \dots, y_1, y_2, \dots$ are Cauchy.

OK: The sequences x_1, x_2, \dots , and y_1, y_2, \dots , are Cauchy.

Good: The sequences (x_i) and (y_i) are Cauchy.

OK: The image of S under f , $f(S) = \{x \mid x \in S\}$, is convex.

Good: The image of S under f , given by $f(S) = \{x \mid x \in S\}$, is convex.

Do not insert superfluous words if the meaning is clear.

Good: Consider the function $f + g + h$, where $f : \mathbf{R}^n \rightarrow \mathbf{R}$, $g : \mathbf{R}^m \rightarrow \mathbf{R}$, and $h : \mathbf{R}^p \rightarrow \mathbf{S}^n$ are closed proper convex.

References. For internal references, use the `label` and `ref` commands. *Never* number or refer to an entity using a specific number, as in “Table 3.” Refer to “Table 1”, “(1)” (an equation), and so on. Only number equations that are important and should be emphasized or that you specifically refer to elsewhere. Do not simply number all equations, and do not number them randomly. A numbered equation draws the reader's attention, so it should be used relatively sparingly. Use `\S` for section references, as in §3.

For external references, you must use `BibTeX`. Use references like nouns or footnotes:

Good: One useful reference is [BV04].

Good: Interested readers may refer to Boyd and Vandenberghe [BV04].

Your BibTeX source must be correct. Unfortunately, many systems that export BibTeX entries export very poor ones. BibTeX ignores the *.bib file's capitalization for articles (but not books). To force capitalization, you must wrap the letter with curly braces. The hyphen (-), en dash (–), and em dash (—) are distinct punctuation marks that serve different purposes. (And the minus sign is different from all three.) When specifying a page range in references, use the en dash. For an example of escaping and using the en dash, look at our BibTeX entry for [Tre08].

Do not italicize English in math mode. Mathematical symbols should be typeset in math mode: write $Ax = b$, not $Ax=b$. This said, subscripts or superscripts that derive from English (or any human language) should not be italicized. For example, write f_{best} , not f_{best} . The exception is subscripts based on a single letter: refer to a point that is the center of some set as x_c , not x_c . Similarly, use commands for special functions: use $\sin(x)$, $\log(x)$, and $\exp(x)$, not $\sin(x)$, $\log(x)$, or $\exp(x)$.

A really heinous example would be the following:

Consider the problem

$$\textit{minimize } f(Ax - b)$$

where x is the optimization variable and A and b are problem data.

Spacing. A blank line ends a paragraph. You shouldn't leave a blank line between an equation and the following text unless you intend the equation to end the paragraph. Write:

```
The image of $$$ under $f$,
\[
  f(S) = \{ f(x) \mid x \in S \},
\]
is convex.
```

Inserting extra blank lines before `\[` or after `\]` will result in bad typesetting. However, the following is fine, since a new paragraph is called for:

```
The image of $$$ under $f$ is defined as
\[
  f(S) = \{ f(x) \mid x \in S \}.
\]
```

We now turn to a different topic.

Using a tab before $f(S)$ inside the equation environment is optional, but helps readability in the source. Similar rules should be followed for other environments like `quote`; see the source of this document.

Use of notation and jargon. The correct and appropriate use of notation and jargon takes time to master. The following should give you a sense of what to think about:

- Don't use the same notation for two different things. Conversely, be consistent about notation for the same thing mentioned in two places: don't say " A_j for $1 \leq j \leq n$ " in one place and " A_i for $i = 1, \dots, n$ " in another, or use two different indices in summations over the same ranges. Formally, the i and j are dummy variables with only that phrase as scope, so these are technically correct, but it will confuse the reader.
- It can be useful to choose names for indices so, for example, i always varies from 1 to m and j always varies from 1 to n (when referencing something like the rows and columns of an $m \times n$ matrix).
- Define all symbols before or near to where you use them. A good rule is that when you first use something, say, X , you should either define it immediately, or within the paragraph. There are a few cases where it is acceptable to define it later, but you must say this explicitly, as in " X is the matrix of activation levels, which we define below."
- A symbol like f refers to a function, while $f(x)$ refers to a function evaluated at a given point. Avoid sloppy writing like "The function $f(x)$ is convex." So-called 'anonymous' functions defined inline are an exception to this rule, as in "the function $x^2 \cos x$ is a counterexample," though this should be used only for simple functions.
- Use typographic naming conventions like capital letters for sets, Greek letters for dual variables, *etc.*
- Try to use mnemonic notation, so x_c for a center point, c for a cost vector, S for a generic set, C for a convex set, B for a ball, and so on. Note that there are conventions about the use of different letters in mathematics that you should not ignore: i , j , and k are usually used for indices, x through z for variables, a through d for constants, *etc.*
- Don't use symbols like \forall , \exists , and \Rightarrow ; use the corresponding words. These symbols are usually appropriate only in formal logic.
- Don't overuse subscripts. If you do not need to assign a subscript to something, don't. For example, if you begin by defining a set as $X = \{x_1, \dots, x_n\}$, then it is going to get annoying to refer to subsets of X , since you will need double subscripts. If possible, it would be better to not name the elements of X and only refer to them when necessary, just as, say, x and y .

Beware especially of any topic in graph theory when it comes to this rule; there are often *many* ways to refer to relevant objects, and finding reasonable notation can often get one half the way to getting a handle on the problem itself.

- Don't assign symbols to concepts that you never refer to, or can easily refer to without:

Bad: Let X be a compact subset of a space Y . If f is a continuous real-valued function over X , it has a minimum over X .

Good: A continuous real-valued function has a minimum over a compact set.

Similarly, do not say “The solution x^* is unique” if you never need to refer to x^* again; simply say that the solution is unique. When you say “The solution x^* is unique” you are both stating a fact *and* entering the symbol x^* into the paper’s symbol table and the reader’s working memory.

Sections. Use structures like `section`, `subsection`, `subsubsection`, and `paragraph` to organize the exposition; never insert manual line breaks or page breaks for this purpose. In particular, do not use line breaks to separate different topics. Do not use symbols in article titles and ideally not even in section headers.

Use capitalization consistently. For section headers, capitalize the first letter, proper nouns, and lowercase everything else, as in this document. For references, write Figure 1, Algorithm 3, Theorem 4, and so on.

Use the right commands. There are certain special commands in \LaTeX for notation that you otherwise might attempt to write in an ad-hoc manner. If you do the latter, the typesetting will be inferior. A few common examples:

- Norms are written $\|x\|$, not $||x||$.
- For set-builder notation, use $\{x \in \mathbf{R} \mid x \geq 0\}$, not $\{x \in \mathbf{R}|x \geq 0\}$.
- Make sure to use `\left` and `\right` when wrapping taller expressions with parentheses, curly braces, brackets, and so on.

Bad: Let

$$x = \operatorname{argmin}_u \left(f(u) + \frac{1}{2} \|u - z\|_2^2 \right).$$

Good: Let

$$x = \operatorname{argmin}_u \left(f(u) + \frac{1}{2} \|u - z\|_2^2 \right).$$

- Use `\ldots` (lower dots) when the dots are surrounded by commas and `\cdots` (center dots) when surrounded by other objects that have full height, as in x_1, x_2, \dots, x_n and $x_1 + x_2 + \cdots + x_n$.

Similarly, make sure to use our conventions for standard mathematical objects for this class; for example, use \mathbf{R} , not \mathbb{R} , to denote the set of real numbers. Use the `\operatorname{argmin}` and `\operatorname{argmax}` commands (as shown above); do not write ‘arg min’, since ‘argmin’ is a single mathematical operator. These and other definitions are provided in `defs.tex`.

Writing optimization problems. In this class, optimization problems can be introduced in the sentence as nouns. For example, write as follows:

Consider the problem

$$\begin{aligned} & \text{minimize} && (1/2)\|Ax - b\|_2^2 + \lambda\|x\|_1 \\ & \text{subject to} && 0 \preceq x \preceq \mathbf{1} \\ & && \|x\|_2 \leq 1, \end{aligned} \tag{1}$$

where $x \in \mathbf{R}^n$ is the optimization variable, and $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$, and $\lambda > 0$ are problem data.

It is important to state which symbols refer to variables and which to problem data.

Be sure to carefully distinguish an optimization problem, an algorithm for solving it, and its optimal value. You *solve* a problem (possibly, using a particular algorithm); an optimization problem *has* an optimal value; and an optimization problem *is* convex, or infeasible. It does not make sense to talk about whether or not a problem converges.

Sentence-ending periods. L^AT_EX assumes all periods followed by a space are sentence-ending periods. Tell it otherwise when that is not the case.

Bad: Let x_1, x_2, \dots, x_n be i.i.d. normal random variables.

Good: Let x_1, x_2, \dots, x_n be i.i.d. normal random variables.

Commas. Know when commas should appear inside or outside math environments:

Bad: Note that $a, b,$ and c are nonnegative.

Good: Note that $a, b,$ and c are nonnegative.

Bad: We conclude that x_1, x_2, \dots, x_n are decreasing.

Good: We conclude that x_1, x_2, \dots, x_n is decreasing.

Good: We conclude that $x_1 > x_2 > \dots > x_n$.

Tables and figures. Generally, do not refer to figures or tables in a way that depend on their placement on the page; give them a `label` and then use `ref` to refer to it. All figures and tables should have captions.

Bad: Properties of several functions are shown in the table below.

Good: Properties of several functions are shown in Table 1.

L^AT_EX in figures. Generally, mathematical symbols embedded in figures should be rendered by L^AT_EX. In Python, matplotlib has native support for L^AT_EX rendering. Otherwise, you can export the image in eps format and use the L^AT_EX package psfrag. See Figure 1 and Figure 2. To use psfrag, compile using

```
$ latex template.tex
$ dvi2pdf template.dvi
```

The psfrag package does not work with pdflatex.

$f(x)$	convex?	differentiable?
$\ x\ _2^2$	Y	Y
$\ x\ _1$	Y	N
$\ x\ _2^3$	N	Y

Table 1: Example table

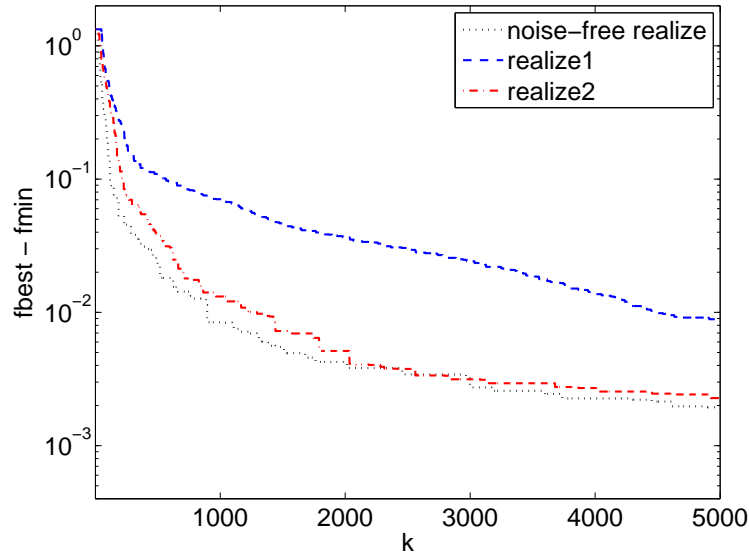


Figure 1: Original image without psfrag.

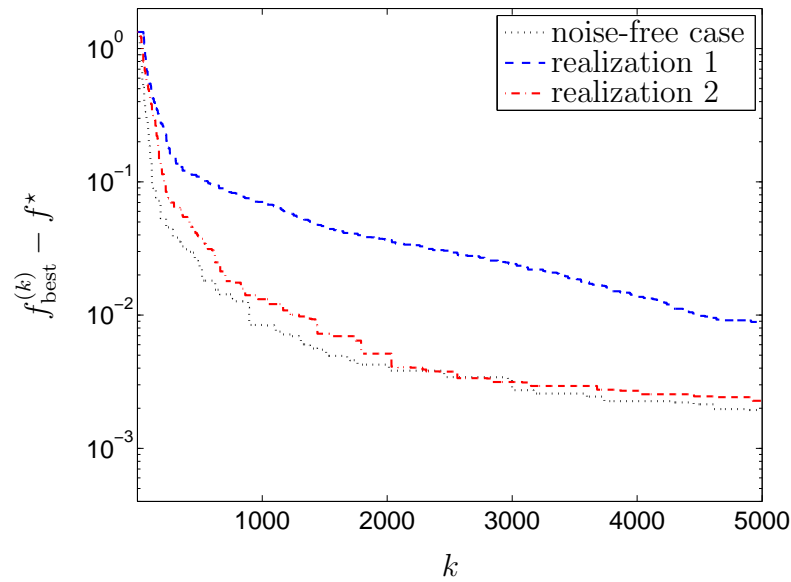


Figure 2: Text replaced with equations using psfrag.

Dialects. Be aware when you are writing in a mathematical dialect, like statistics, machine learning, signal processing, control, finance, vision, information theory, and so on. Unless your intended audience is only from this one field, try to avoid using dialect; try to write in such a way that a general reader with a good understanding of basic mathematics can understand what you are saying. If you use terms like CVAR, hinge loss, Bode plot, or IIR filter, you're speaking dialect. It's very easy to define these things so a person with knowledge of basic mathematics can understand. Similarly, use standard variable notation unless otherwise needed: x for variables, A for matrices, and so on. Don't write a whole paper about solving a linear system you call $\Xi\beta = \chi$ unless you really need to.

No rule is absolute. Break any of these rules rather than write anything nasty.

2 Source code

In this example, we show how to include code in a document.

```
rand('seed',0);
randn('seed',0);

m = 1000;
n = 3000;
p = 10/n;

x0 = sprandn(n,1,p);
A = randn(m,n);
A = A*spdiags(1./sqrt(sum(A.^2))',0,n,n);
b = A*x0 + sqrt(0.001)*randn(m,1);

tic
cvx_begin
    variable x(n)
    minimize(0.5*sum_square(A*x - b) + norm(x,1))
cvx_end
toc

prox_f = @(v,lambda) (1/(1 + lambda))*(v - b) + b;
prox_g = @(v,lambda) max(0, v - lambda) - max(0, -v - lambda);
x_admm = admm(prox_f, prox_g, A);
```

Here, we could now include some discussion about the code, include plots it generates, and so on. Use your judgment about whether to start a new page before displaying code, as it is helpful to be able to display all the code on a single page. For very short snippets of code (just a few lines), it can help readability to wrap it in a `quote` environment.

3 Content

Your report should use the following structure:

1. Begin with a brief paragraph that provides background, perhaps in somewhat field-specific language. Describe the problem you hope to address in general terms. For example: “We consider the problem of reconstructing an image, given a blurred noisy version of it.” Ideally, this opening should be snappy and interesting, otherwise you will lose the intended audience.
2. Transition (quickly) into a mathematical problem statement that anyone with general mathematical training can look at and understand, without any domain knowledge.
3. Outline the general approach. As a general comment here, it is of the *utmost importance* not to mix together the problem statement and the method you intend to use to solve instances of that problem. You should cover these in separate paragraphs, if not separate sections or subsections. This is a key conceptual point, not just about writing. Note that the approach is not merely the optimization algorithm you will eventually use. For example, your approach to solving a nonconvex problem may be to produce a particular convex relaxation whose solution will give a heuristic solution for the original problem. Here, the relaxation is as much a part of the approach as the algorithm you use to solve the relaxation.
4. If relevant, discuss properties of your overall approach, such as convergence, rate of convergence, implementation details, and so on. This could be as simple as saying that you use CVX, or something more involved for, say, a distributed ADMM implementation. Make sure you explain what machine and software you used.
5. If relevant, present some numerical experiments. If the focus is on an algorithm, it is important to thoroughly describe the problem data and settings of any algorithm parameters (step sizes, stopping criteria, and so on). You should include number of iterations to convergence, wall-clock time to convergence, residuals at convergence, and any other related metrics. If appropriate, compare your method to some relevant baseline, like CVX. If the focus is on solving a practical problem, describe the problem instance (or instances) solved, where the data came from, and so on. The general problem statement should have been stated before, so when you present the numerical example, you should not need to define a new problem. For example:

We now consider a specific instance of the index tracking problem (13). We have $n = 100$ assets, with daily return data from January 2005 to January 2012, all obtained from Google Finance. The transaction cost parameter κ was set to 0.002, *i.e.*, 20 basis points, which is consistent with typical transaction costs in practice.

Typically you would describe the actual *results* in a different paragraph.

6. If your original problem is in a particular field, make sure to try to pause at various points in the exposition to provide interpretations of the mathematical properties, approach, or numerical results back in the original domain.
7. Briefly conclude and summarize the high-level points.

Acknowledgments

This is an example of an unnumbered section.

References

- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [CSRL01] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.
- [CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.
- [Eva10] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2010.
- [Gol80] H. Goldstein. *Classical Mechanics*. Addison-Wesley Publishing Company, 2nd edition, 1980.
- [Hal70] P. R. Halmos. How to write mathematics. *L'Enseignement Mathématique*, 16:123–152, 1970.
- [HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- [KLR89] D. E. Knuth, T. Larrabee, and P. M. Roberts. *Mathematical Writing*, volume 14 of *MAA notes*. Mathematical Association of America, 1989.
- [Knu73] D. E. Knuth. *The Art of Computer Programming, Volume I: Fundamental Algorithms*. Addison-Wesley, 2nd edition, 1973.
- [Rud76] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill Book Co., third edition, 1976.
- [Sip01] M. Sipser. *Introduction to the Theory of Computation*. Thomson Course Technology, 2001.
- [SW57] W. Strunk and E. White. *The Elements of Style*. Macmillan, 1957.
- [Tre08] L. N. Trefethen. Is Gauss quadrature better than Clenshaw-Curtis? *SIAM Review*, 50:67–87, 2008.