

Introduction to L^AT_EX

Gregory D. Landweber

October 4, 2007

1 A brief history of typesetting

1.1 Handwritten manuscripts

Copying books and other manuscripts by hand takes time and produces very few copies. That's good for out-of-work monks, but bad for Sophocles when his plays burned with the library at Alexandria. To this day, the Hebrew Torah is still copied by hand onto a scroll of parchment.

1.2 The printing press (1436)

Good for Gutenberg and the protestant reformation. Prompts monks to pursue other things, like beer. Gives rise to the typesetting industry, putting together printed manuscripts one character at a time.

1.3 The typewriter (late 1800's)

Good for people with bad handwriting. Bad for carpal tunnels.

1.4 The Linotype machine (1880's)

Automates typesetting a whole line o' type (get it!) at a time. Great for newspapers and font designers.

1.5 The laser printer (1970's)

Ushers in the age of digital typesetting, and with it the tradition of publishers mangling authors' manuscripts in the typesetting process.

1.6 $\text{T}_{\text{E}}\text{X}$ (1978)

A markup language allowing the author to digitally typeset his own manuscripts. Created by the computer scientist Donald Knuth while on sabbatical from Stanford, as a response to the mangled galley proofs of the second edition of *The Art of Computer Programming* (as well as his disgust with the typesetting in AMS journals). Described in *The T_EXbook* (1984).

1.7 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}_{\text{E}}\text{X}$ (1982–1985)

A macro package written by Michael Spivak and described in *The Joy of T_EX*. Simplifies typesetting various mathematical constructions, and used by the American Mathematical Society for its journals. Superseded by $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

1.8 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (1983–1985)

A high-level language consisting of an extensive set of macros that sits on top of TeX and reorganizes its functionality. Created by Leslie Lamport, who is now at Microsoft Research. Described in *L^AT_EX: A document preparation system* (1985).

1.9 WYSIWYG (1984)

Apple introduces the Macintosh with its graphical user interface and “What You See Is What You Get” word processing. Microsoft releases Word. Popularizes the Geneva, New York, and Chicago fonts, which look awful when printed on a dot matrix printer.

1.10 Desktop publishing (1985)

Apple introduces the LaserWriter, a “personal” laser printer costing \$7000 and the first to use Adobe’s PostScript page description language. Aldus introduces PageMaker, a graphical layout program.

1.11 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (1990)

The AMS ports $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}_{\text{E}}\text{X}$ ’s functionality to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

1.12 Greg’s Senior Project (1992)

Greg wastes his time writing his senior project using Microsoft Word 5.1 and the Expressionist equation editor. Forgive him. He was young and stupid, and nobody gave him an “Introduction to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ” talk.

1.13 \LaTeX 2 ϵ (1994)

The latest update to \LaTeX , which reorganizes LaTeX rather significantly. A compatibility mode is provided to process old \LaTeX .

1.14 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\LaTeX$ 1.2 (1995)

The latest update to $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\LaTeX$, ported to \LaTeX 2 ϵ .

1.15 \TeX 3.141592 (2002)

The latest version of \TeX . No bugs found in five years!

1.16 Greg's Senior Project Revisited (2005)

Greg rewrites his senior project using $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\LaTeX$. It looks so much better now!

1.17 \LaTeX 3 (future)

In the works for 18 years and counting.

2 What is \TeX ?

\TeX is both a markup language (like HTML) for digital typesetting, as well as the program which takes a text file written in \TeX , usually with a file name ending in “.tex”, and converts it to a form which can be directly typeset, such as DVI or more recently PDF. For many years, you needed to further convert a DVI file to PostScript in order to print it using a third party program. Ugh!

Since digital typesetting is a boring repetitive process, Knuth decided to automate it. Instead of focusing on user experience like WYSIWYG word processors, \TeX focuses on the finished product, producing output that rivals the best professional typesetters. \TeX is programmed with all known rules of conventional and technical typesetting, taking care of every detail, such as margins, formatting, spacing, and even hyphenation. Back in the early 1980's, \TeX consistently produced better output than today's best word processors.

\LaTeX takes the basic idea of \TeX and refines it. Here are some of its features:

- Introduces document classes, such as book, report, and article, which have different formatting. These are like CSS files used with HTML.
- Automatically numbers and resolves references to equations, page numbers, sections, and bibliography entries.
- Automatically generates tables of contents.

- So much more that I take for granted, not having used raw $\text{T}_{\text{E}}\text{X}$ in ages. Hmm. I wonder how hard it is to do this bulleted list in raw $\text{T}_{\text{E}}\text{X}$?

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ adds functionality to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ for use in mathematics. Sure, you can typeset plenty of mathematics without $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, and I have colleagues who do. However, $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ is very slick and makes life much easier. Its features include:

- Better control of displayed equation alignment, spacing, and numbering.
- Better handling of matrices, roots, arrows, and fraction-like constructions.
- Lets you create your own operator names, like \sin and \log , which typeset correctly (before $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, this was a pain).
- Basic commutative diagrams
- Standardized format for theorems and proofs.
- Adds a whole new set of math fonts and symbols.

3 How do I get $\text{T}_{\text{E}}\text{X}$?

At its heart, $\text{T}_{\text{E}}\text{X}$ is a command line tool, used mostly on Unix variants. However, there are versions of $\text{T}_{\text{E}}\text{X}$ for Macs and Windows. Most versions include, in addition to the $\text{T}_{\text{E}}\text{X}$ engine itself, a graphical text editor for creating the $\text{T}_{\text{E}}\text{X}$ source code, together with a viewer to see the resulting DVI, PostScript, or PDF output. Most importantly, the distribution of $\text{T}_{\text{E}}\text{X}$ that you use should be free. If it isn't, make sure that it offers you something special that is really worth the money (one old variant of $\text{T}_{\text{E}}\text{X}$ for the Mac OS had a feature where it would automatically $\text{T}_{\text{E}}\text{X}$ your document as you typed).

On the Mac, I recommend $\text{T}_{\text{E}}\text{X}$ shop, a graphical front end to the Unix $\text{T}_{\text{E}}\text{X}$ command created by Dick Koch at the University of Oregon. It is available as part of the $\text{MacT}_{\text{E}}\text{X}$ distribution. One nice thing about $\text{T}_{\text{E}}\text{X}$ shop is that it automatically generates PDF files, bypassing the whole DVI/Postscript step (although that is still available for certain $\text{T}_{\text{E}}\text{X}$ constructions that need it). $\text{T}_{\text{E}}\text{X}$ shop is open source and won an Apple Design Award. I contributed a tiny bit of the code for handling macros and fixing a few bugs. The URL for $\text{T}_{\text{E}}\text{X}$ shop is: <http://www.uoregon.edu/~koch/texshop/>.

4 $\text{T}_{\text{E}}\text{X}$ basics

In practice, you learn $\text{T}_{\text{E}}\text{X}$ by using it. Sure, you can read a $\text{T}_{\text{E}}\text{X}$ manual, but there is no substitute for needing to write a fraction and looking up how to do it. Your best bet is to look at other people's $\text{T}_{\text{E}}\text{X}$ files. One great place to find $\text{T}_{\text{E}}\text{X}$ files is the arXiv at <http://www.arxiv.org/>. In particular, you should not try to start writing a $\text{T}_{\text{E}}\text{X}$ file with

a blank page. \TeX shop has templates for different kinds of documents which set up all the complicated stuff in the preamble for you, so that you need only worry about the text.

\TeX follows professional typesetting conventions. In particular, it automatically knows about ligatures, like ff, fi, fl (without ligatures, you get ff, fi, fl, which are just so ugly). It has smart quotes, but you have to specify them yourself. You use a combination of single forward and single backward quotes as follows: ‘this is single quotes’ and “this is double quotes, with two single quote characters”. \TeX supports various accents, such as é è ä ç. \TeX also supports various types of dash. We have the small hy-phen, a longer “n”-dash for use in ranges, like 1–10, and the longest “m”-dash for use—as you see often in books—for interrupting a thought.

In \TeX , you separate paragraphs by a blank line. Be careful about blank lines. It is easy to put them in your text accidentally when trying to space things out,

but \TeX recognizes them and acts accordingly. If you have some suspect indentation or extra vertical space that should not be there, look for extra blank lines. If you really want to space out your thoughts without starting a new paragraph, you can use \TeX comments, which start with a % symbol. Good \TeX editors like \TeX shop highlight comments in a different color so that you know they are not part of the text.

\TeX takes care of the word wrapping, margin justification, paragraph spacing, and everything else that you would otherwise tweak yourself in a word process. The algorithms that \TeX uses are quite clever. All of the spacing is flexible, and \TeX adjusts it to make the entire paper look as good as possible. It tries to avoid widows and orphans, i.e., lines with only one word on them, or paragraphs that spill over with just one line on a new page. It keeps track of a measure of “badness”, and will make the spacing on one page slightly worse if that improves the spacing on another page significantly. Try that in Microsoft Word!

In general, you want to let \TeX fiddle with the fonts, but you do have several standard choices, including *italics*, **sans-serif**, SMALL CAPS, and **teletype**.

\LaTeX handles bibliographic citations with ease. For instance, I am very fond of Grätzer’s book, which is reference [1].

5 Typesetting mathematics

When you want to typeset mathematics, you tell \TeX to enter “math mode”. Everything is typeset differently in math mode than in regular text. Symbols are written in math italics, and two letters adjacent to each other are typeset as the product of two variables. There are different syntaxes for entering math mode, but then one I prefer is using \$ (the dollar-sign—which becomes a Yen symbol in Japanese \TeX). To write Einstein’s famous equation $e = mc^2$, you type

```
$e = mc^{2}$
```

With a single \$ dollar sign, the mathematics is typeset as part of the sentence inline with the rest of the text. Notice the use of the caret (shift-6) for the superscript and the braces

{,} around the 2. Subscripts work the same way with an underscore (shift-hyphen). Here is an expression that mixes the two, $\ddot{x}^i = \gamma_{jk}^i \dot{x}^j \dot{x}^k$, written in \TeX as

```
 $\ddot{x}^i = \gamma_{jk}^i \dot{x}^j \dot{x}^k$ 
```

Here, note the standard syntax that \TeX commands start with a backslash and enclose their parameters in curly braces. All of the greek letters are spelled out in full.

Equations that are too large to fit inline with the rest of the text can be displayed centered on a separate line in `displaymath` mode. This is delimited by a double dollar sign `$$`. For instance, we could write

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

by \TeX ing

```
 $\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$ 
```

For equations that you want to refer back to later, instead of using `$$`, you can use the `\mathcal{AMS-}\LaTeXequation` environment, as in

$$\int_a^b f(t)g'(t)dt = f(b)g(b) - f(a)g(a) - \int_a^b f'(t)g(t)dt. \tag{5.1} \quad \boxed{\text{integration}}$$

Written as

```
\begin{equation}\label{integration-by-parts}
  \int_{a}^{b} f(t)g'(t)dt = f(b)g(b) - f(a)g(a) - \int_{a}^{b} f'(t)g(t)dt.
\end{equation}
```

Notice here the use of \LaTeX begin-end blocks, and also that in math mode, the apostrophe character `'` becomes a prime `'`.

Note that \LaTeX provides the equation number automatically, and you can refer to it by the label you specified as `(5.1)`, written as

```
\eqref{integration-by-parts}
```

this-section

You can also put labels anywhere you want and refer to them as in Section 5.

```
\label{this-section}
Section~\ref{this-section}
```

The `~` symbol here is a non-breaking space, to avoid the section number appearing on a separate line from the word before it. If you want these cross references to work, you need to run your source through \LaTeX twice. The first time it creates a table of the labels, and the second time it inserts the correct cross references into the output. When I wrote this file, I added the *showkeys* package, which makes cross referencing easier by showing the labels in

the margins. For the finished product, I would comment out the *showkeys* package line in the preamble.

Other $\mathcal{AMS-L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ commands let you align multiple equations nicely, as in

$$x + y = 1 \tag{5.2}$$

$$x - y = 3 \tag{5.3}$$

written as

```
\begin{align}
x + y &= 1 \\
x - y &= 3
\end{align}
```

Here, you use the ampersand & as the alignment character, and a double backslash to denote the end of the line. Note that you get equation numbers even if you don't specify a label, not that you could do much with them. If you want to suppress the label, replace *align* with *align**. (Similarly, using *equation** puts no label on a single equation, which is functionally the same as the double dollar sign \$\$.)

Other handy environments are *matrix*, *bmatrix*, and *pmatrix*, for matrices without delimiter, with square brackets, and with parentheses, as in

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

written as

```
$$
\begin{matrix}1&2\\3&4\end{matrix}
\begin{bmatrix}1&2\\3&4\end{bmatrix}
\begin{pmatrix}1&2\\3&4\end{pmatrix}
$$
```

Again, this uses the alignment character & to separate the columns, and the double backslash to separate the rows.

Finally, if you have a big expression that needs larger than normal parentheses or brackets, you can use the following

```
$$\sum_{i=1}^n a_i \left( \sum_{j=1}^m b_j \right) .$$
```

to generate

$$\sum_{i=1}^n a_i \left(\sum_{j=1}^m b_j \right).$$

The *left* and *right* commands tell $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ to measure how big the content is between them and adjust the size of the parentheses accordingly. This works with square brackets and curly brackets as well.

References

- [1] George Grätzer, *Math into L^AT_EX*. (I ordered it for the Bard library)

This is the L^AT_EX manual which I keep by my side when writing papers (except that I cannot find it since I moved). It has the best treatment of the $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX packages.

- [2] Helmut Kopka and Patrick W. Daly, *Guide to L^AT_EX*. (I ordered it for the Bard library)

A highly regarded L^AT_EX manual.

- [3] Frank Mittelbach and Michel Goossens, *The L^AT_EX Companion*.

For advanced L^AT_EX users, this describes some of the most useful of the zillions of add on packages to L^AT_EX. If there is something you want to do that L^AT_EX doesn't already do, odds are that someone has written a package to do it, and it is described here.

Many books on T_EX and L^AT_EX describe obsolete versions. Avoid Knuth's *The T_EX book*, Spivak's *The Joy of T_EX*, and the first edition of Lamport's *L^AT_EX: a document preparation system*. Always be sure to get the latest edition, since many of the commands described in older editions of books may have been changed in later versions of L^AT_EX.