

Writing Mathematics with L^AT_EX

Dr. Kevin P. Lee

1 Introduction

What is L^AT_EX¹?

Typesetting mathematical formulas can be a difficult and tedious. Most commercial word processing packages, like Word and WordPerfect, are not designed for extensive mathematical, scientific, and technical writing. L^AT_EX is a computer system for typesetting technical papers and documents. It is used widely by scientists and mathematicians to produce books, papers, and articles.

This document is meant to help you get started with L^AT_EX. It is by no means a complete manual. To go beyond the basics of L^AT_EX, you should read a more comprehensive guide such as *L^AT_EX: A Document Preparation System: User's Guide and Reference Manual* by Leslie Lamport, or *Math into L^AT_EX: An Introduction to L^AT_EX 2_ε and A_MS-L^AT_EX 1.2* by George Grätzer. These books, and others like it, are available in the Purdue Calumet library. Other resources for L^AT_EX are available on the world-wide web (check out the Mathematical Writing web page).

Why Use L^AT_EX?

Though learning to use L^AT_EX may be difficult at first, you will find that once you have learned it, you will be able to compose mathematical papers with greater speed. You will be able to typeset formulas and equations more easily than with other word processing programs. L^AT_EX is designed to typeset equations and formulas with minimal fuss and professional quality.

For these reasons, L^AT_EX is now the universal standard for producing mathematical and scientific papers and articles. L^AT_EX runs on many different types of computers and is

¹When you talk to mathematicians and scientists, you will often hear the word T_EX used instead of L^AT_EX. L^AT_EX is based upon T_EX. When you are using L^AT_EX, you are using T_EX.

compatible with a wide range of printers. Mathematicians exchange papers and submit articles to journals over the internet using \LaTeX .

The Philosophy of \LaTeX

Most modern word processing programs adhere to the idea WYSIWYG, or “What you see is what you get.” When you type a paper in Word or WordPerfect, you need to worry about the fonts, the font size, the spacing between lines, indentations, and many other aspects of the *look* of your paper.

\LaTeX is different. With \LaTeX , the computer is responsible for all of the formatting. The writer is responsible for telling \LaTeX the *logical structure* of her or his paper. Suppose that we want to have a chapter titled “The Journey Begins”. In a WYSIWYG word processing program, we would have to tell the computer to use a particular font and a particular font size for each of the words in the chapter title. With \LaTeX , we only need to indicate that the words “The Journey Begins” are part of a chapter title. The computer then figures out what will be the appropriate font, the font size, the correct margins, the correct spacing, and any other bits of formatting.

2 Getting Started

Setting It Up

There are many versions of \LaTeX which can be installed on a computer. On the internet, you can find free and low-cost shareware implementations of \LaTeX available for download. For those who like bells and whistles, there are also more expensive commercial implementations of \LaTeX . The computers in the Math Computing Lab already have \LaTeX installed. (For more information on using \TeX at Purdue Calumet, read *Using \TeX and \LaTeX at Purdue Calumet*.)

When you write papers with \LaTeX , you will not actually be typing anything into the \LaTeX program itself. Instead, the input is typed into a text editor, which is a no-frills word processing program. In a text editor, you cannot change fonts, font sizes, font styles, spacings, indentations, or any of the visual appearance of the text. (With \LaTeX , this won't matter anyway.) Most Macs and Windows machines come with text editors already installed (SimpleText on the Macintosh, Notepad on Windows) and there are more powerful text editors available online for free or as low-cost shareware. It is possible to use a traditional word processing program, like Word and WordPerfect, as a text editor. However, you must save the file as text. (You can do this by selecting “Save As...” in the “File” menu and selecting “Text only” as the file format.)

The text source file will typically be saved with the suffix `.tex`. You can then run a \TeX program (there are many different such programs) on the `.tex` source file. This will produce a `.dvi` file.² (If you began with a file `myfile.tex`, you will have a new file called `myfile.dvi`.) The `.dvi` file is the formatted document with the beautiful-looking formulas and equations. The `.dvi` file can be viewed with a DVI-viewer. (Just like there are many \TeX programs, there are many different DVI-viewers.) If you want to make some changes, you will need to edit the `.tex` source file (with the text editor) and then run it through the \TeX program again. Frequently, you will also encounter `.aux` and `.log` files as well. These files are just records of what occurred while the `.tex` file was being converted into a `.dvi` file; they can be ignored. (If you learn more sophisticated \LaTeX techniques, you will come to encounter other types of \LaTeX files with suffixes like `.bib`, `.idx`, and `.toc`.)

Many \TeX implementations combine the \TeX -compiler and the DVI-previewer into one program. A few \TeX implementations combine the text editor with the \TeX -compiler as well. The specifics of running \TeX will vary between different platforms and implementations.

Beginning With a Source File

Instead of writing a `.tex` source file from scratch, you may find it easier to modify an already existing file. On the Mathematical Writing web page, there are two such files you can use: `longtemp.tex` and `shrttemp.tex`. The file `longtemp.tex` contains extensive commentary and has many examples of \LaTeX syntax, while `shrttemp.tex` is a more basic bare-bones template. You may also wish to look at the file `sample2e.tex`, which is obviously a sample \LaTeX file.

Elements of a Source File

If you look at the sample files closely, you should see that there are some common elements to all \LaTeX source files. Some particular characters are special in \LaTeX . The `%` character indicates the beginning of a comment. \TeX will ignore everything after a `%` character. The backslash, `\`, indicates a \TeX command.

Every \LaTeX `.tex` file starts with a `\documentclass` command, like:

```
\documentclass{article}
```

or

² \TeX will balk at producing a `.dvi` file if there is an error in the source `.tex` file. The most common errors are typos and missing a `{` or `}` symbol.

```
\documentclass{report}
```

(By the way, \TeX distinguishes between uppercase and lowercase letters in commands. Don't get sloppy with this!) For more details about the `\documentclass` command, consult one of the \LaTeX reference books mentioned in the introduction. Or you can read the comments contained in the file `longtemp.tex`.

The `\documentclass` command is typically followed by several `\usepackage` commands. \LaTeX is a highly flexible system. Extra commands can be added to \LaTeX using the `\usepackage` command. For instance,

```
\usepackage{amsmath}
```

tells \LaTeX to recognize some additional commands particularly useful for mathematical typesetting. (The “ams” stands for American Mathematical Society.) The command

```
\usepackage{graphicx}
```

adds some special commands to \LaTeX which allow you to use imported graphics within a document. Again, you may wish to consult a \LaTeX reference or to examine the file `longtemp.tex` for more details.

After the `\documentclass` and `\usepackage` commands, each \LaTeX source file also contains the commands:

```
\begin{document}
```

and

```
\end{document}
```

All of the actual content of your document will be located between these two commands.

3 Text

The `.tex` source file consists of regular text without any formatting. There are ten special characters:

```
# $ % & _ { } ~ ^ \
```

As mentioned above, `%` indicates the beginning of a \TeX comment, while `\` indicates a \TeX command. The other symbols above also indicate special \TeX actions. If you need one of the first seven special symbols in a document, you can type `\#` for “#”, `\$` for “\$”, etc.

\LaTeX ignores all tabs, returns, and extra spaces. So

```
This      is
      some sample
      text.
```

and

```
This is some sample text.
```

will give the same result:

```
This is some sample text.
```

A blank line in the source indicates to \LaTeX the end of a paragraph. If you want to force \LaTeX to begin a new line, type `\\` at the end of the line. So

```
Twinkle twinkle little star, \\ How I wonder what you are.
```

will result in

```
Twinkle twinkle little star,
How I wonder what you are.
```

One of the most powerful features of \LaTeX is its ability to organize documents. You can organize a \LaTeX document by using the `\section` and `\subsection` commands. So to start a section titled “Why Dogs Are Better Than Cats”, type into the source file

```
\section{Why Dogs Are Better Than Cats}
```

More details on sectioning commands can be found in \LaTeX reference books and in the comments located in `longtemp.tex`.

4 Working with Formulas

Writing mathematical formulas with \LaTeX is easier if the `amsmath` and `amssymb` packages are loaded. So for the rest of this chapter, I will assume that the lines

```
\usepackage{amsmath}
\usepackage{amssymb}
```

are contained at the beginning of the document.

Formulas are contained between $\$$ signs. When you type

```
So, the graph of  $y = x^2$  is symmetrical about the  $y$ -axis.
```

\LaTeX recognizes that $y = x^2$ is part of a mathematical formula. The result is

```
So, the graph of  $y = x^2$  is symmetrical about the  $y$ -axis.
```

Notice how \LaTeX has used an italic type style for the mathematical variables.

You can also enclose formulas and expressions between $\$$'s. This will place the formula on its own separate line. The line

```
So, the graph of  $y = x^2$  is symmetrical about the  $y$ -axis.
```

within the source file will produce in the output

```
So, the graph of
```

$$y = x^2$$

```
is symmetrical about the  $y$ -axis.
```

You can also easily create numbered equations with the commands

```
\begin{equation}
```

and

```
\end{equation}
```

Typing

```
\begin{equation}
y = 3x^2 + 2x - 1
\end{equation}
```

within the `.tex` source file will give

$$y = 3x^2 + 2x - 1 \tag{1}$$

for the output. \LaTeX will automatically number the equations. If you insert another numbered equation earlier in a document, \LaTeX will automatically renumber all of the later equations. For \LaTeX can perform this trick, whenever you have numbered equations in a document, you will need to run your `.tex` source file twice through the \TeX program.

To refer back to a numbered equation, you will need to use the `\label` command somewhere between the `\begin{equation}` and `\end{equation}` commands. For example, you can type into the `.tex` file

```
We will need to use the formula,  
\begin{equation}  
f(x) = x^{3} + 2x - 5. \label{anylabel}  
\end{equation}
```

You can of course use anything instead of `anylabel` (preferably, something that is meaningful and easy to remember). The resulting output won't look any different:

We will need to use the formula,

$$f(x) = x^3 + 2x - 5. \tag{2}$$

But you can now use the `\ref` command to refer back to the equation. The input

```
Thus, from equation \ref{anylabel}, we can see that the equation  
$f(x) = g(x)$ has no solution.
```

will give

Thus, from equation 2, we can see that the equation $f(x) = g(x)$ has no solution.

Again, for \LaTeX to get all the numbering correct, you will need to run the source file through the \TeX program twice.

You should use a \LaTeX reference for a more complete description of the typesetting features and commands contained in \LaTeX . The *$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{E}\mathcal{T}\mathcal{E}\mathcal{X}$ Version 1.2 User's Guide* and *Math into $\mathcal{E}\mathcal{T}\mathcal{E}\mathcal{X}$: An Introduction to $\mathcal{E}\mathcal{T}\mathcal{E}\mathcal{X}$ 2 ϵ and $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{E}\mathcal{T}\mathcal{E}\mathcal{X}$ 1.2* by George Grätzer are particularly helpful. (Check out the appendices for tables of mathematical symbols. The *$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{E}\mathcal{T}\mathcal{E}\mathcal{X}$ Version 1.2 User's Guide* contains useful information about breaking up formulas over several lines.) The table on the next page shows some examples which exhibit some of the most commonly used \LaTeX constructions.

L ^A T _E X Source	Resulting Output
<code>4 \times 3 + 10 \div 5 - 3 = 33/3</code>	$4 \times 3 + 10 \div 5 - 3 = 33/3$
<code>-1 < \lvert 2x + 3 \rvert \leq 5</code>	$-1 < 2x + 3 \leq 5$
<code>\frac{3}{2}</code> <code>\geq 2\left(z + \frac{5}{2}\right)</code> <code>> \frac{-1}{2}</code>	$\frac{3}{2} \geq 2\left(z + \frac{5}{2}\right) > \frac{-1}{2}$ (Occasionally, parentheses may be the wrong size. The <code>\left</code> and <code>\right</code> command choose the right size of parentheses. There are similar constructions, like <code>\left\lvert</code> and <code>\right\rvert</code> for the vertical lines denoting absolute value.)
<code>\sqrt[3]{8} + \sqrt{49} = 81^{1/2}</code>	$\sqrt[3]{8} + \sqrt{49} = 81^{1/2}$
<code>f(x) = a_{0} + a_{1}x + a_{2}x^{2} + \cdots</code>	$f(x) = a_0 + a_1x + a_2x^2 + \cdots$
<code>f(x) = \begin{cases}</code> <code>-x^{2} & \text{if } \\$x < 0\$,}</code> <code>x^{3} & \text{if } \\$x \geq 0\$}</code> <code>\end{cases}</code>	$f(x) = \begin{cases} -x^2 & \text{if } x < 0, \\ x^3 & \text{if } x \geq 0 \end{cases}$
<code>\{1, 2, 3\} \subset (\{1, 4\} \cap \{1, 5\})</code> <code>\cup \{2, 3, 4\}</code>	$\{1, 2, 3\} \subset (\{1, 4\} \cap \{1, 5\}) \cup \{2, 3, 4\}$
<code>x_{1}, x_{2}, x_{3}, \ldots \in \mathbb{R}</code>	$x_1, x_2, x_3, \dots \in \mathbb{R}$
<code>-10 \notin [3, 7]</code>	$-10 \notin [3, 7]$
<code>\sin(\theta + \pi/2) = \cos \theta</code>	$\sin(\theta + \pi/2) = \cos \theta$
<code>\ln(a + b) \neq \ln a + \ln b</code>	$\ln(a + b) \neq \ln a + \ln b$
<code>\lim_{n \rightarrow \infty}</code> <code>\sum_{i=1}^n f(x_i) \Delta x =</code> <code>\int_a^b f(x) dx</code>	$\lim_{n \rightarrow \infty} \sum_{i=1}^n f(x_i) \Delta x = \int_a^b f(x) dx$
<code>f'(a) = \lim_{h \rightarrow 0}</code> <code>\frac{f(a+h) - f(a)}{h}</code>	$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}$
<code>\vec{v} \cdot \vec{w} = \lvert \vec{v} \rvert \lvert \vec{w} \rvert \cos \theta</code>	$\vec{v} \cdot \vec{w} = \ \vec{v}\ \ \vec{w}\ \cos \theta$
<code>\text{grad} f(x, y) =</code> <code>\frac{\partial f(x, y)}{\partial x} \vec{i} +</code> <code>\frac{\partial f(x, y)}{\partial y} \vec{j}</code> (A <code>\vec{i}</code> command would place an arrow over a dotted "i". This is unattractive. The commands <code>\imath</code> and <code>\jmath</code> produce produce better looking results.)	$\text{grad } f(x, y) = \frac{\partial f(x, y)}{\partial x} \vec{i} + \frac{\partial f(x, y)}{\partial y} \vec{j}$ (A <code>\vec{i}</code> command would place an arrow over a dotted "i". This is unattractive. The commands <code>\imath</code> and <code>\jmath</code> produce produce better looking results.)
<code>\angle BAC < 90^\circ</code>	$\angle BAC < 90^\circ$
<code>\ell_1 \parallel \ell_2</code>	$\ell_1 \parallel \ell_2$
<code>\ell_1 \perp \ell_2</code>	$\ell_1 \perp \ell_2$
<code>\triangle ABC \cong \triangle DEF</code>	$\triangle ABC \cong \triangle DEF$
<code>\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix},</code> <code>\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix},</code> <code>\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}</code>	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$

5 Pictures and Graphics

To include a graphic in a \LaTeX document, the graphic needs to be saved as an Encapsulated Postscript file (`.eps`). Some printer drivers can generate such files. Also, some software packages (like Maple) can generate these types of files.

In the source file, you will need to make sure that you include the line

```
\usepackage{graphicx}
```

toward the beginning.

Suppose that you have saved your graphic file as `mygraphic.eps`. Make sure that the file is located in the same folder as your `.tex` source file. Then you can place the picture inside the document by writing the line

```
\includegraphics{mygraphic.eps}
```

You can change the size of the graphic in several ways. Writing

```
\includegraphics[scale = 2]{mygraphic.eps}
```

makes the graphic `mygraphic.eps` twice as large (while `scale = 0.5` makes the graphic one-half of its original size). Typing

```
\includegraphics[width = 1.5in]{mygraphic.eps}
```

scales the graphic so that it is 1.5 inches wide, while

```
\includegraphics[height = 1.5in]{mygraphic.eps}
```

scales the graphic so that it is 1.5 inches high. (Note that there are no spaces between `1.5` and `in` in the above examples. This is important to the \LaTeX syntax.)

You should be aware that some DVI-viewers will not properly display included graphics files. However, even if the DVI-viewer does not work, you should be able to print the file out correctly.

6 Additional Comments

One of the most powerful features of \LaTeX is the ability to easily create your own macros. For instance, in \LaTeX , there is no shortcut to write \mathbb{R} . You could define `\R` to be a macro for this symbol by writing in the source file the line

```
\newcommand{\R}{\mathbb{R}}
```

Afterwards, instead of writing $\$x \in \mathbb{R}\$,$ you could write $\$x \in \R\$$ to get the expression $x \in \mathbb{R}$.

For another example, suppose that you get tired of writing the sequence of commands for the absolute value symbols. You could include in the source file

```
\newcommand{\abs}[1]{\left\lvert #1 \right\rvert}
```

The [1] indicates that there will be one argument for the new `\abs` command. The #1 shows where this argument should appear in the macro. Now, instead of typing $\$\left\lvert x + y \right\rvert\$,$ you could type $\$\abs{x + y}\$$ to get $|x + y|$ in the output.

Even more complicated macros can be constructed in \LaTeX . You can even include macros within other macros. (Be careful with this! You may wish to consult a \LaTeX manual if you wish to utilize a lot of macros.)

There are many other features of \LaTeX and \TeX which have not even been mentioned here. In \LaTeX , there are methods for automatically generating tables, lists, bibliographies, indexes, and tables of contents. You do not have to learn *all* of \LaTeX to start using \LaTeX . But I hope that as you become more experienced with \LaTeX , you too will learn to appreciate its elegance and power.