

L^AT_EX TUTORIAL

ZANE MARTIN

1. THE WHAT AND WHY OF L^AT_EX

L^AT_EX is a powerful and relatively straightforward mathematical publishing tool that is specifically suited for the needs of mathematicians. It provides the means to express mathematical statements, formulae, and symbols simply and intuitively, producing results which are standardized, legible, and well-organized. This tutorial introduces you to the basic concepts and conventions of L^AT_EX to allow you to begin writing simple documents like homeworks and exams. A sample homework has been included to show you an appropriate formatting option, though once you become more literate in L^AT_EX you'll be able to forge your own style. The best way to learn is to experiment with this code to get a feel for the program. In addition, there are a number of helpful resources available online.

To download L^AT_EX go to oit.williams.edu/software. Windows users can select MiKTeX, or the more feature-rich TeXnicCenter, and Mac users should choose TexShop. The downloads are both free and the links are trusted, though the download may be slow.

1.1. Acknowledgments. This tutorial was completed under the supervision of Cesar E. Silva, Professor of Mathematics at Williams College, as part of the Williams College Summer Science program. Support for the project was provided by the Bronfman Science Center. Crucial in providing inspiration for the content and structure of this document, as well as a starting point for the project, was “Introduction to LaTeX”, a LaTeX document by Daniel Kane completed September 1, 2011.

2. BASIC FORMATTING AND TYPING

2.1. Commands and Formatting. In L^AT_EX, what you type in the code and what is displayed as a .pdf file are rarely the same. L^AT_EX accepts regular text in many cases, but we can also code for certain actions or symbols using specific code words preceded by a backslash “\”. These code words are case-sensitive, and are the foundation of effectively using L^AT_EX (for example, getting this fancy font to say LaTeX we type in \LaTeX). The code accompanying this document gives you the framework needed to set up a document and installs the required packages.

Once you have your document ready, there are a number of ways you can divide it up in L^AT_EX. You can use the “\begin{ }” command to begin sections, exercises, proofs, theorems,

lemmas, etc, simply by placing these words within the curly brackets. So to start an exercise we type “\begin{exercise}.”

Exercise 1. *Everything that we type after that appears here in our exercise. \LaTeX is smart enough to number them automatically. To leave we type “\end{exercise}”.*

Or if we want try something else...

Proof. Typing in “\begin{proof}” gets us here. I did not have to type in the italicized proof, all I did was begin the proof and then type “Typing in “\begin{proof}”...” The rest is automatic. To escape we type “\end{proof}” (the little box is included free of charge) \square

2.2. Typing and Manipulating Fonts. Once you’re in a section, exercise, or proof, denoted by the commands, write as you would in any word processor. If you want to start a new line, leave an empty line in your code. To insert a new paragraph, type the command “\” followed by an empty line. To prevent the new paragraph from indenting (the default) type the command “\noindent” at the beginning of the line.

To change the style of your fonts, you can use a number of commands. **\bf write in boldface**, **\it writes in italics**, and **\rm** brings it back to Roman (even in an equation, see Section 3). You can change the size of text with the following commands: **\tiny**, **\small**, **\normalsize**, **\large**, and **\Large**.

3. MATH MODE AND EQUATIONS

To enter “math mode” simply place a “\$” before and after the mathematical expression, symbol, command, etc. This changes the way \LaTeX formats text. Thus, rather than simply having some variable x , we can express our variable as x (in the code $\$x\$$). Similarly $1+1=2$ in math mode looks like $1 + 1 = 2$. Be sure that any equations, sets, numbers, or variables are in math mode, and be aware that many of the commands addressed below only work in math mode.

To set an equation on its own line, use “[” at the beginning of the equation, and “\]” at the end. The effect looks like

$$7 + 3 = 10.$$

Note that if we try to add text in an equation (or in math mode for that matter) we get

This doesn't look great.

To make it look better we can use “\rm” to change the font and “\” whenever we wish to have a space. So we get

This looks better.

There is no need to use “\$” when in an equation. Remember to place period or commas before the end of the equation (]) to prevent formatting errors.

In many cases we will want to place several equations one after the other. We can do this a number of ways. First, we can place them each in their own set of brackets (\[equation \]) to get

$$\begin{aligned}
 3x + 2 &= 8 \\
 3x &= 6 \\
 x &= 2
 \end{aligned}$$

To clean this up a bit we can use an equation array. Using the command “`\begin{eqnarray*}`”, then using our double backslash to make spaces between equations, we can get

$$\begin{eqnarray*}
 3x + 2 &= 8 \\
 3x &= 6 \\
 x &= 2
 \end{eqnarray*}$$

Finally, we can use “&” to line up elements of our array right above one another. By writing “&=” in place of “=”, we get the finished product

$$\begin{eqnarray*}
 3x + 2 &= 8 \\
 3x &= 6 \\
 x &= 2
 \end{eqnarray*}$$

4. SYMBOLS

4.1. Operators. The basic operators and relations “=”, “+”, “/”, and “−” are simply typed on the keyboard and require no special commands. The same is true for “<” and “>”, although if we forget to place them in “math mode” they become “i” and “j” respectively. The following are some other helpful operator and relation symbols

Operators and Relations			
<code>\cdot</code>	·	<code>\equiv</code>	≡
<code>\times</code>	×	<code>\ast</code>	*
<code>\div</code>	÷	<code>\cap</code>	∩
<code>\le</code>	≤	<code>\odot</code>	⊙
<code>\ge</code>	≥	<code>\centerdot</code>	•
<code>\approx</code>	≈	<code>\cup</code>	∪
<code>\oplus</code>	⊕	<code>\setminus</code>	\
<code>\sqcap</code>	⊓	<code>\sqcup</code>	⊔

For inequalities like \neq or $\not\leq$, we can say that something is “not” by using the command “`\not`” before the command we wish to enter. So \neq looks like “`\not=`”. We can extend this past inequalities to say something is “not in” a set or a group, to say a set is “not a subset” of another set etc., all simply by adding “`\not`” before our desired commands.

4.2. Greek Letters and Special Letters. Another very useful set of symbols provided by L^AT_EX are the letters of the Greek alphabet. To use these, we simply type a “`\`” followed by the name of the Greek letter. For capital letters, simply capitalize first letter of the command. While aleph is not Greek letters, it has been included in the following table. In addition, capital Greek letters which resemble their Latin counterparts (a capital alpha is an “A”) have been excluded

Greek Letters									
<code>\alpha</code>	α	<code>\theta</code>	θ	<code>\pi</code>	π	<code>\psi</code>	ψ	<code>\Pi</code>	Π
<code>\beta</code>	β	<code>\iota</code>	ι	<code>\rho</code>	ρ	<code>\omega</code>	ω	<code>\Psi</code>	Ψ
<code>\gamma</code>	γ	<code>\kappa</code>	κ	<code>\sigma</code>	σ	<code>\aleph</code>	\aleph	<code>\Sigma</code>	Σ
<code>\delta</code>	δ	<code>\lambda</code>	λ	<code>\tau</code>	τ	<code>\Gamma</code>	Γ	<code>\Theta</code>	Θ
<code>\epsilon</code>	ϵ	<code>\mu</code>	μ	<code>\upsilon</code>	υ	<code>\Delta</code>	Δ	<code>\Upsilon</code>	Υ
<code>\zeta</code>	ζ	<code>\nu</code>	ν	<code>\phi</code>	ϕ	<code>\Lambda</code>	Λ	<code>\Xi</code>	Ξ
<code>\eta</code>	η	<code>\xi</code>	ξ	<code>\chi</code>	χ	<code>\Phi</code>	Φ	<code>\Omega</code>	Ω

Another useful set of letters used in math are those used to express important groups of numbers like the natural numbers or the integers. To denote these we write a backslash followed by the capital letter we wish to use. So \mathbb{N} is written “`\N`”. We can also use the command “`\mathbb{}`” to express the same thing (so \mathbb{R} is written “`\mathbb{R}`”).

4.3. Other Important Symbols. The following table provides some important symbols you will likely need. This list is *by no means* exhaustive. It should also be noted that programs like WinEdit or TeXnic Center have drop-down menus which provide many symbols and will automatically input the code. Finally, there are many websites online, so finding a specific symbol is usually not very difficult.

Other Symbols					
<code>\partial</code>	∂	<code>\subset</code>	\subset	<code>\leftarrow</code>	\leftarrow
<code>\infty</code>	∞	<code>\subseteq</code>	\subseteq	<code>\rightarrow</code>	\rightarrow
<code>\triangle</code>	\triangle	<code>\subsetneq</code>	\subsetneq	<code>\Leftarrow</code>	\Leftarrow
<code>\exists</code>	\exists	<code>\supset</code>	\supset	<code>\Rightarrow</code>	\Rightarrow
<code>\emptyset</code>	\emptyset	<code>\supseteq</code>	\supseteq	<code>\Leftrightarrow</code>	\Leftrightarrow
<code>\in</code>	\in	<code>\supsetneq</code>	\supsetneq	<code>\leftrightharpoons</code>	\leftrightharpoons
<code>\prime</code>	\prime	<code>\forall</code>	\forall	<code>\lnot</code>	\lnot

In addition to these symbols, there are a few which will change their size depending on whether they are used in an equation or in text. The main ones we are concerned with are \int (`\int`), \bigcup (`\bigcup`), \bigcap (`\bigcap`), \sum (`\sum`). Note that if we write $\int x + 2 dx$ in our text, it is different than

$$\int x + 2 dx.$$

We do not need to change anything in the code, \LaTeX is able to do this automatically. All we do is place the second one in an equation using “`\[`”

4.4. Punctuation. \LaTeX offers a number of types of punctuation, many of which you will use with regularity. The standard `()`, `[]`, and `||` are all entered using the keyboard symbols without any type of code, as are periods, commas, semicolons, etc. The only tricky bracket is `{}`, which is often used in commands like “`\begin{theorem}`” or in subscripts and superscripts (see Section 4). These normally do not show up in text or math mode, so if we want them to appear we write “`\{`” and “`\}`.” In general placing a “`\`” before a symbol makes it visible in the final output.

The only other tricky punctuation is quotation marks. If we place a word in quotes as we are used to, it appears like this: "quotes". Note that the first set of quotation marks is backwards. We need to use the ‘ symbol (upper left hand corner on a QWERTY keyboard) twice in place of the first quotation mark. The final result is “quotes.”

To avoid overly small parentheticals such as

$$\left(\frac{1}{1+\frac{1}{2}}\right).$$

we use the commands “\left(” and “\right)” (we can use any parenthetical in place of “()”). So writing “\left(” instead of just “(” we get

$$\left(\frac{1}{1+\frac{1}{2}}\right)^n.$$

A few other useful punctuation marks are summarized in the following table

Punctuation Marks			
<code>\dots</code>	...	<code>\dotsb</code>	...
<code>\vdots</code>	⋮	<code>\ddots</code>	⋱
<code>\tilde{x}</code>	\tilde{x}	<code>\hat{x}</code>	\hat{x}
<code>\ddot{x}</code>	\ddot{x}	<code>\dot{x}</code>	\dot{x}

The difference between “\dots” and “\dotsb” is that “\dotsb” are higher off the bottom of a line. Thus, when listing items with commas (1, 2, ..., n) we use “\dots.” When listing items using a binary operator (the “b” in “\dotsb”) such as 1 + 2 + ... + n, we use “\dotsb.”

5. ADVANCED FORMATTING OF EXPRESSIONS AND EQUATIONS

5.1. Subscripts and Superscripts. To do a subscript simply use the underscore key “_”. So to write a_1 we type in “a_1.” If we want something a little more complicated like a_{n+1} , we use curly brackets to write “a_{n+1}”. Without the curly brackets, our expression codes as $a_n + 1$. A superscript behaves exactly the same, using the “^” key. We can combine superscripts and subscripts to express many of the ideas we will need in mathematical papers, solutions, etc.

$$n_{x^{y+2}+14}^{\alpha_0+\rho_{i_j}}, \int_0^{x+h}, \sum_{i=0}^{\infty}, \lim_{n \rightarrow -\infty}$$

5.2. Roots, Overlines, and Fractions. To take the square root of something we write “\sqrt{expression}” and place the entire expression taken in the brackets ($\sqrt{\text{expression}}$). In general we can type in “\sqrt[n]{ }” to take the the n^{th} root of something ($\sqrt[n]{\text{something}}$). Overlines are very similar; we use the command “\overline{ }” ($\overline{\text{overline}}$).

For writing fractions, we use the command “`\frac`” followed by two sets of curly brackets containing the numerator and denominator respectively. So the code “`\frac{frac}{tion}`” will produce

$$\frac{frac}{tion}$$

To express the binomial coefficient, we format the exact same way but use the command “`\binom`.”

5.3. Special Operators. The following list of operators are already named by L^AT_EX. This means that in math mode, these commands should be used rather than simply writing out the operator. For example, if we want to say $\sin(x)$ we need to type “`\sin(x)`.” Otherwise it looks like $sin(x)$: the product of the variables s , i , n , and x . In each case the code and the displayed text are the same, we use the coding for convenience and convention.

<code>\arccos</code>	<code>\arcsin</code>	<code>\arctan</code>	<code>\cos</code>	<code>\cosh</code>	<code>\cot</code>	<code>\coth</code>
<code>\csc</code>	<code>\deg</code>	<code>\det</code>	<code>\dim</code>	<code>\exp</code>	<code>\gcd</code>	<code>\inf</code>
<code>\ker</code>	<code>\lim</code>	<code>\liminf</code>	<code>\limsup</code>	<code>\ln</code>	<code>\log</code>	<code>\max</code>
<code>\min</code>	<code>\sec</code>	<code>\sin</code>	<code>\sinh</code>	<code>\sup</code>	<code>\tan</code>	<code>\tanh</code>

This should give you all you need to get started. The following is a sample homework; please note the formatting and the use of several of the symbols and concept that have been covered in this tutorial.

Sample Homework

LaTeX 101, September 8, 1755

My Name

Exercise 2 (chapter.section.problem#). *Include a complete statement of the problem.*

Proof. Use a proof to answer the question. Follow the instructions provided in the tutorial and please ensure your work is clear and all important steps are shown. Theorems, lemmas, propositions, etc. should all be properly cited, as should references to other problems or exercises. Make sure you proofread your work carefully before handing it in. \square

Exercise 3 (2.22.89.). *Show that if $x^2 + 2 - 5 = 6$, then $x = 3$.*

Proof. We use algebra on the given equation to show the following:

$$\begin{aligned} x^2 + 2 - 5 &= 6 \\ x^2 - 3 &= 6 && \text{add 2 and } -5 \\ x^2 - 3(+3) &= 6(+3) = 9 && \text{add 3 to both sides} \\ \sqrt{x^2} &= \sqrt{9} = 3 && \text{take the square root of both sides} \end{aligned}$$

Thus, we conclude that $x = 3$. \square

Exercise 4 (1.26.91.). *Calculate the sum of the numbers 1 through 5.*

Proof. We express the problem in summation notation as

$$\sum_{n=1}^5 n = 1 + 2 + 3 + 4 + 5.$$

Using mental math and checking our work with a calculator we find that $1 + 2 + 3 + 4 + 5 = 6 + 4 + 5 = 10 + 5 = 15$. Thus,

$$\sum_{n=1}^5 n = 15$$

Exercise 5 (3.14.15). *Calculate the integral from 1 to 4 of the constant function $x = 3$.*

Proof. We express the problem as

$$\int_1^4 3 \, dx.$$

Using the integration laws we proved on Homework X Exercise Y we know that

$$\int_1^4 3 \, dx = 3x \Big|_1^4 = 3(4) - 3(1) = 12 - 3 = 9$$

Therefore $\int_1^4 3 \, dx = 9$. \square

S.U BOX 1347 PARESKY CENTER, WILLIAMS COLLEGE, WILLIAMSTOWN, MA 01267
E-mail address: `zkm1@williams.edu`