

Writing a Paper with L^AT_EX

Joe Struss

Jan. 28, 2011

Contents

1	Creating a Paper—	
	The Basics	3
1.1	Getting Started	3
1.2	The Document Wrapper	3
1.3	The Document	4
2	Making Changes to a Paper	7
2.1	Common Documentclass Options	7
2.2	Style Alterations to a Documentclass	7
2.3	Helpful Document Commands	9
2.4	Additional L ^A T _E X Packages	10
3	Other Parts of the Paper	12
3.1	Title Page	12
3.2	Bibliography	12
3.3	Mathematics	13
3.4	Dividing your paper into parts	13
4	Mathematics Power	14
4.1	Lining up on a decimal point	16
4.2	Under-/Over-Brackets and Parentheses	17
4.3	Hypothesis and Bold Lowercase Greek Characters	18
4.4	Putting a Box around a Parbox with Math Formula	19
4.5	The Amsmath and Latexsym Packages	20
5	Accents and Writing in Español with the Babel Package	21
5.1	General L ^A T _E X Accents and Babel	21
5.2	Using Spanish Babel: An Extended Example	22

6	Creating a Presentation with \LaTeX	24
6.1	The Slides Documentclass	24
6.2	The Beamer Documentclass	26
6.3	3-D Objects	27
A	Font Types and Styles	28
B	PDF Files and .EPS Graphics	29
C	General \LaTeX Information	30
D	Basic \LaTeX on a UNIX System	31
	References	33

Chapter 1

Creating a Paper— The Basics

1.1 Getting Started

If you are using L^AT_EX on a Windows or Macintosh, you would normally create your document in the editor provided with the TeXnicCenter, TeXWorks or TeXShop software then compile your .tex document using the PDFL^AT_EX compiler by pressing a Build or Compile button. If there are any errors, you would need to correct them and then re-compile the document. If the document compiles correctly, you should simply be able to view the resulting .pdf file or in most cases it will be displayed to you. See [4] for more information.

1.2 The Document Wrapper

The first thing you need to do in a L^AT_EX document is to tell L^AT_EX the type of document that are creating— this is known as choosing a *documentclass*. You do this with the `\documentclass{class}` command. There are five basic document classes: **book**, **report**, **article**, **slides** and **letter**. The report and article classes are good for short papers while book is good for longer papers. Letter and slides are more specific purpose documentclasses. Letter is used for communications and memos while the slides class is used to produce presentational slides.

The only other thing that L^AT_EX really needs to know is where your document begins and ends. Basically you put a wrapper around your document with the `\begin{document}` and `\end{document}` commands.

```
\documentclass{report}
\begin{document}
\end{document}
```

1.3 The Document

1.3.1 Paragraphs

Between the `\begin{document}` and `\end{document}` commands, you put in the paragraphs for your document. Simply insert one or more blank lines which signals to L^AT_EX to start a new paragraph. L^AT_EX ignores the way you format your sentences on a line and just looks for a paragraph break and then strings together all the lines that compose a particular paragraph:

```
\documentclass{report}
\begin{document}
Here is the first sentence of
my document.
Here is my second sentence.

After a paragraph break, I start
a new sentence.
I can put just
one
word
per line if I want to do so.
It makes no difference at all to LATEX.
\end{document}
```

1.3.2 Special Characters

Watch out for the characters: `&`, `#`, `$`, `%`, `_`, `^`, `~`, `{`, and `}`. These are special reserved characters in L^AT_EX. If you want to use one of these characters in a document, you will need to put a `\` in front of them. For example: `\$` will put a `$` into a document. Additionally, you need to put `{ }` after `^` and `~` or these characters get squeezed too much by the characters around them like this `\~{}`.

The other special characters are: `+`, `=`, `|`, `<`, `>` and `\`. These special characters are only available in math mode so you have to put `$`'s around them in order to use them. For example: `+$` will put a `+` into a document.

Additionally, since the `\` is also the symbol for a reserved character, if you actually want to use a `\` character in a document- besides just being in math mode you will also need to add the word `backslash` after the `\`. This ends up looking like this in a document: `\backslash` which is a lot of work just to get a `\`. You're actually encouraged in \LaTeX to use the `\newcommand` command to make an alias for a group of \LaTeX commands that you often use and that's what people often do for a `\`. For instance, you could enter in the preamble: `\newcommand{\bs}{\ensuremath{\backslash}}` to use `\bs` for backslash in your document.

1.3.3 Quotes

\LaTeX uses angular ‘quotes’ so beginning quotes like this: ‘ ‘ look different than ending quotes like this: ’ ’ . The left single quote is normally next to the `1` key on the keyboard while the right single quote key is the normal quote (or apostrophe key) next to the `;` key.

Never use the double-quote key when creating a \LaTeX document unless you want an umlaut or some other non-english character. Use two beginning quotes “followed by two ending quotes” instead or your ”quotes” may look funny as they will both be angled in the same direction.

1.3.4 Dashes

There are three types of dashes available in \LaTeX . For a word dash, like in x-ray, use a single hyphen `-`. For a range dash, like in 1–10, use two hyphens in a row `--`. And for a sentence dash, like this—, use three hyphens in a row `---`. The hyphens that are showing here are a little more spaced out than normal so that they can be seen.

1.3.5 Emphasis and Bolding Characters

Use `\emph{word}` command to *emphasize* a word or phrase in your document. Use `\textbf{word}` to bold a word or phrase in your document.

There are a variety of other font changes that you can make in \LaTeX . These additional font styles are listed in Appendix [A](#). Here are some examples.

To italicize the word *dog*, you would normally use the command in a “closed form” for italic shape which is `\textit{dog}`. You could also use the declaration in an “open form” which would be `{\itshape dog }`. The `\itshape` in an “open form” would more commonly be used to italicize an

entire paragraph. The paragraph would start with `\itshape` and end with `\upshape`.

Type style changes only occur in an “open form” known as a declaration. For instance, to create elephant in a very small font— you would use `{\tiny elephant }`. You could also use `\tiny elephant \normalsize` which would get you the same result. You could make the text in a paragraph a little smaller by using `\small` at the start of the paragraph and then `\normalsize` at the end it— like what was done for this paragraph.

There are nine different text sizes, three additional shapes, one additional series and two additional font families that come standard with \LaTeX and are dependent on the point size with which you begin your document. Standard point sizes are 10 point (default), 11pt and 12pt. Play around with changing \LaTeX font shape, series, sizes and styles until you are comfortable with making changes to them.

1.3.6 Full Example Document

Here is an example document that shows paragraphs, some special characters, quotes, dashes, emphasis and bolding. Try compiling this sample document with \LaTeX yourself:

```
\documentclass{report}
\begin{document}
And in conclusion— — —
this ultimately showed that
the direct line of support for
the ‘fifth’ division
was \emph{paramount} in
achieving victory.

The 2$+$2 protocol was
used in sending coded messages
during this \textbf{decisive} conflict.
\end{document}
```

Which looks like the following when processed by \LaTeX :

And in conclusion— this ultimately showed that the direct line of support for the “fifth” division was *paramount* in achieving victory.

The 2+2 protocol was used in sending coded messages during this **decisive** conflict.

Chapter 2

Making Changes to a Paper

2.1 Common Documentclass Options

2.1.1 Altering the Base Font Size

You can add options in []'s to the documentclass line in order to make base level changes to a document class. For instance, a L^AT_EX document defaults to a 10pt font size. To change the font size to a base 11pt or 12pt font, you would add [11pt] or [12pt] to the document line as follows:

```
\documentclass[11pt]{report}
```

2.1.2 Making a Twocolumn Paper

Another option you can add is [twocolumn] which when added to a 12pt base font article class document would look like this:

```
\documentclass[12pt,twocolumn]{article}
```

Once this option is in place, your paper will start in two columns. To switch back to a one column style in a two column document enter `\onecolumn`. Once you are in the one column form, you can get back to two column form by entering `\twocolumn`. **Note:** You will get a page break whenever you use `\onecolumn` or `\twocolumn`.

2.2 Style Alterations to a Documentclass

You put in the preamble area of your L^AT_EX document those changes or alterations that affect your entire document or paper. The preamble area is between the `\documentclass` line and the `\begin{document}` line.


```
\documentclass{report}
⇒Preamble stuff goes here.
\begin{document}
```

2.2.1 Altering the Margins

Changing the Side Margins

The easiest way to change the side margins in \LaTeX is to alter the length of a full line of text on the page. Just enter:

```
\setlength{\textwidth}{6 in}
```

in the preamble of your paper.

The `textwidth` can be specified in inches (in) or centimeters (cm) when followed by the appropriate unit of measure.

Changing the Top and Bottom Margins

Just like changing the side margins, the easiest way to change a top or bottom margin in \LaTeX is to alter the length of the height of the text on the page. Just enter:

```
\setlength{\textheight}{9.5 in}
```

in the preamble of your paper.

The `textheight` can be specified in inches (in) or centimeters (cm) when followed by the appropriate unit of measure.

Making the Margins as Narrow as Possible

If you just want just about as big of a standard page as you can get enter:

```
\usepackage{fullpage}
```

in your preamble. The `fullpage` package automatically sets the smallest margins for a standard size page.

2.2.2 Altering Document Spacing

You can alter the document spacing by adding the following line to the preamble of your document or paper:

```
\renewcommand{\baselinestretch}{Factor}
```

where *Factor* is any decimal number. The default factor = 1 so a factor of 1.5 would be a 50% increase in the amount of space between lines in your document. Common factor values are 1 for single-spacing, 1.25 for traditional one and one-half spacing and 1.667 for traditional double-spacing.

Another way to alter document spacing is to use the `setspace` package:

```
\usepackage{setspace}
```

The `setspace` package defines four commands for you: `\singlespacing`, `\doublespacing`, `\onehalfspacing`, and `\setstretch{size}`. You can then use these commands in your document to alter your document spacing.

2.3 Helpful Document Commands

Here are some additional \LaTeX commands for altering your document:

<code>\newpage</code>	\implies Forces \LaTeX to go to a new page.
<code>\newline</code>	\implies Starts a new line within a paragraph.
<code>\indent</code>	\implies Used at the start of a paragraph \implies to force paragraph indentation.
<code>\noindent</code>	\implies Used at the start of a paragraph \implies to stop paragraph indentation.
<code>\footnote{Footnote here.}</code>	\implies Creates a footnote.
<code>\centerline{Line to Center.}</code>	\implies Center a single line.
<code>\hspace{ 1 in}</code>	\implies Creates 1 inch of horizontal space \implies within a paragraph.
<code>\vspace{- .1 in}</code>	\implies Creates -.1 inch of vertical space \implies before the next paragraph.

The `quotation` environment indents a section of text an equal amount from both sides of the page:

```
\begin{quotation}
The sands of time will denote well
the undertakings and accomplishments
that we do or do not fulfill here today.
\end{quotation}
```

The sands of time will denote well the undertakings and accomplishments that we do or do not fulfill here today.

The `center` environment allows you to produce one or more lines of centered text on a page:

```
\begin{center}
First line of text.\\
A second line which is longer.\\
The last line of text.\\
\end{center}
```

First line of text.
A second line which is longer.
The last line of text.

2.4 Additional L^AT_EX Packages

2.4.1 Font Packages

Getting different fonts to load into L^AT_EX has been difficult to do in the past. To make this easier to accomplish, the following font packages were created for use in the preamble of your L^AT_EX document:

times, newcent, palatino, bookman and helvet

So to use the times package, one would enter `\usepackage{times}` in the preamble of a document and then the fonts used in your document would be from the Times Roman font family. *Note:* Not all font packages are available in all versions of L^AT_EX.

2.4.2 Color Package

The color package in a PDF document can be used to add a splash of color here or there to make your document a little more interesting and appealing to the reader. To use the color package in a PDF document, add the following line to the preamble of your document:

```
\usepackage[usenames,dvipsnames,pdftex]{color}
```

You can then use the following commands in your document: `\color{clr}`, `\textcolor{clr}{TEXT}`, `\colorbox{clr}{TEXT}`, `\pagecolor{clr}`, and also `\fcolorbox{line-clr}{clr}{TEXT}`; to change the standard color, `\textcolor` and `\pagecolor` within your document.

The standard document colors are: black, yellow, red, blue, magenta, green and cyan. The additional colors that you can use are: GreenYellow, Yellow, Goldenrod, Dandelion, Apricot, Peach, Melon, YellowOrange, Orange, BurntOrange, Bittersweet, RedOrange, Mahogany, Maroon, BrickRed, Red, OrangeRed, RubineRed, WildStrawberry, Salmon, CarnationPink, Magenta, VioletRed, Rhodamine, Mulberry, RedViolet, Fuchsia, Lavender, Thistle, Orchid, DarkOrchid, Purple, Plum, Violet, RoyalPurple, BlueViolet, Periwinkle, CadetBlue, CornflowerBlue, MidnightBlue, NavyBlue, RoyalBlue, Blue, Cerulean, Cyan, ProcessBlue, SkyBlue, Turquoise, TealBlue, Aquamarine, BlueGreen, Emerald, JungleGreen, SeaGreen, Green, ForestGreen, PineGreen, LimeGreen, YellowGreen, SpringGreen, OliveGreen, RawSienna, Sepia, Brown, Tan and Gray. The complete list in color can be found at: <http://en.wikibooks.org/wiki/LaTeX/Colors>.

Here's a simple text color example:

```
Hello. \color{red} Good-bye. \color{black} Hello again.
```

which would produce:

Hello. **Good-bye.** Hello again.

2.4.3 Graphicx Package

The graphicx package is the standard package which most people use to add a graphic or two to a \LaTeX document. If you are producing a PDF document with \LaTeX then you are limited to using common web graphic types like .jpg, .gif, .pdf and .png for use in your document. To use the graphicx package, add the following to the preamble of your document:

```
\usepackage[pdftex]{graphicx}
```

Then within your document to add your graphic enter:

```
\includegraphics{graphicname.type}
```

Here's an example of putting a graphic into a document. To put a graphic named sprhub.jpg into your document, after you added the graphicx package to the preamble, you would add the following line to your document:

```
\centerline{\includegraphics{sprhub.jpg}}
```

which would produce something like this:



2.4.4 Hyperref Package

The hyperref package adds hypertext links in your PDF document between your Table of Contents, bibliographies and page-references. The hyperref package has a lot of options but can be used in your preamble in a very plain way:

```
\usepackage[pdftex]{hyperref}
```

or here are the options that were used in the preamble of this document:

```
\usepackage[pdftex,hypertextnames=false,linktocpage=true]{hyperref}  
\hypersetup{colorlinks=true,linkcolor=blue}
```

Chapter 3

Other Parts of the Paper

3.1 Title Page

To add a title page to your paper, just add the following lines to your document just after the `\begin{document}` line:

```
\title{Title of Your Document}  
\author{Your O. Name}  
\date{July 4, 1996}  
\maketitle
```

You can leave out the `\date` command and \LaTeX will automatically put in today's date. Additionally you can break a title into multiple lines by using `\\`'s and you can have more than one author's name inside of `\author` if you connect the names with the `\and` command. The title page that \LaTeX creates is somewhat different for each individual document class. To add a "Table of Contents", place `\tableofcontents` after the `\maketitle` command.

3.2 Bibliography

To add a bibliography to your paper with citations you can reference in your document, add the following to the end of your document:

```
\begin{thebibliography}{9}  
\bibitem{label1} Joe Struss. Wanderer.  
Addison–Wesley–Scott Co., Inc., NY, NY, 1984.  
\bibitem{label2} Sam Anders. CEIA— the book.  
Nick/Time Publishing, NY, NY, 1994.  
\end{thebibliography}
```

Now if I wanted to reference my `label1` citation in a paper, I would use the `\cite{label1}` command and add something like “For further information see `\cite{label1}` and `\cite{label2}`” to my document. When this is processed by \LaTeX it comes out like “For further information see [1] or [2]”.

3.3 Mathematics

Mathematics is what \LaTeX does best! A `$` enters you into a simple math mode in \LaTeX while the next `$` will bring you back out again. Here are some basic things you can do while in math mode:

Use `^` to superscript: `A^{23}` $\implies A^{23}$

Use `_` to subscript: `B_{b3}` $\implies B_{b3}$

Use `\frac{top}{bottom}` to make a fraction: `$$\frac{34}{5}$$` $\implies \frac{34}{5}$

Use `\sqrt` to create a square root: `$$\sqrt{42}$$` $\implies \sqrt{42}$

Do: `$$\sum_{i=2}^n b_i$$` to create $\implies \sum_{i=2}^n b_i$

Do: `$$\int_a^c f(x) dx$$` to create $\implies \int_a^c f(x) dx$

There are also lots of special symbols in math mode including a full line of greek symbols like `\beta` β and `\psi` ψ . More information on math mode is available in Chapter 4 or in any standard \LaTeX manual.

3.4 Dividing your paper into parts

The following sectioning commands are available in \LaTeX to formally divide your paper up into separate parts or units:

```

\part      \subsection  \paragraph
\chapter   \subsubsection \subparagraph
\section

```

So if you wanted to start a new chapter named ‘Overview’ you would enter: `\chapter{Overview}`. Normally, you start at the chapter level and work your way down through section, subsection, subsubsection... from there.

Warning: Some document classes do not support all levels of sectioning. For instance, the article document class does not allow either the part or chapter sectioning levels.

Once you have established sections, you can use `\label{nlabel}` and `\ref{nlabel}` to reference your sections. Just put a `\label` command directly after a section that you would like to label and then use `\ref` to reference that label. For instance, I labeled the mathematics section of this chapter with a `\label{math}` command— so I can now use the `\ref{math}` command to say see Section 3.3 for some basic mathematics information.

Chapter 4

Mathematics Power

Mathematics is at the heart of what $\text{T}_{\text{E}}\text{X}$ and $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ do best! Below are some examples of math modes that are available in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. For more complete mathematical information, see the source books listed after the Appendix.

4.0.1 *In-text* mathematics

Use $\$ \dots \$$ or $\backslash(\dots\backslash)$ to create mathematics within a line of text, like this: $\$A^{43}\$$ (which produces A^{43}) by using $\$$ to get into math mode and then another $\$$ to exit math mode. While in math mode, you can use \wedge to produce superscripts and $_$ to produce subscripts but if you have more than one character as a superscript or subscript then you have to enclose your superscript or subscript in braces $\{\}$. Here are some example:

Variable $\$D^7\$$ shows	\implies	Variable D^7 shows
Here $\$Y_a\$$ relates	\implies	Here Y_a relates
This shows: $\$A^{43}\$$	\implies	This shows: A^{43}
Here $\$B_{150}\$$ is true.	\implies	Here B_{150} is true.

4.0.2 *Display* mathematics

Use $\backslash[\dots \backslash]$ to create mathematics that are set apart from your textual material. For instance, you can use $\backslashfrac{\text{num}}{\text{denom}}$ to create a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ large fraction. So an *in-text* fraction, would look like this $\frac{43x}{1698}$, while a display math fraction would look like this:

$$\frac{43x}{1698}$$

Here are some more examples that use superscripts, subscripts, `\frac`, `\sqrt` (square root $\sqrt{42}$), and `\ldots` (ellipsis ...):

$$\backslash [\backslash \text{frac}\{A^{\{123\}} - 7\}\{T_5\} \backslash] \quad \Longrightarrow \quad \frac{A^{123} - 7}{T_5}$$

$$\backslash [\backslash \text{frac}\{1\}\{\sqrt{75}\backslash \text{ldots}\sqrt{123}\} \backslash] \quad \Longrightarrow \quad \frac{1}{\sqrt{75} \dots \sqrt{123}}$$

4.0.3 Mathematical symbols and fonts

There are a large variety of mathematical symbols available in math mode in $\text{T}_\text{E}_\text{X}$ and $\text{L}^\text{A}_\text{T}_\text{E}_\text{X}$. Listed here are just a few standard examples:

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	θ	<code>\theta</code>
Σ	<code>\Sigma</code>	Φ	<code>\Phi</code>	Ω	<code>\Omega</code>	Ψ	<code>\Psi</code>
\dagger	<code>\dagger</code>	\odot	<code>\odot</code>	\star	<code>\star</code>	\circ	<code>\circ</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\subset	<code>\subset</code>	\supset	<code>\supset</code>
\dots	<code>\ldots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>
\equiv	<code>\equiv</code>	\approx	<code>\approx</code>	\neq	<code>\neq</code>	\doteq	<code>\doteq</code>
\leftarrow	<code>\leftarrow</code>	\rightarrow	<code>\rightarrow</code>	\Leftarrow	<code>\Leftarrow</code>	\Rightarrow	<code>\Rightarrow</code>
\triangle	<code>\triangle</code>	∞	<code>\infty</code>	\backslash	<code>\backslash</code>	\exists	<code>\exists</code>
\sum	<code>\sum</code>	\int	<code>\int</code>	\prod	<code>\prod</code>	\coprod	<code>\coprod</code>
\odot	<code>\bigodot</code>	\otimes	<code>\bigotimes</code>	\cup	<code>\bigcup</code>	\sqcup	<code>\bigsqcup</code>

You can put a slash through a symbol by putting a `\not` before it. Some additional mathematical symbols (like \rightsquigarrow and \boxtimes) are available by using the *latexsym* package. See any standard $\text{L}^\text{A}_\text{T}_\text{E}_\text{X}$ book (like [1] or [2]) for a complete list of mathematical symbols available in $\text{L}^\text{A}_\text{T}_\text{E}_\text{X}$. $\text{L}^\text{A}_\text{T}_\text{E}_\text{X}$ also provides the following commands for changing type styles in math mode:

italic: $3i\pi$	<code>\mathit{3i\pi}</code>	roman: $3i\pi$	<code>\mathrm{3i\pi}</code>
sans serif: $3i\pi$	<code>\mathsf{3i\pi}</code>	typewriter: $3i\pi$	<code>\mathtt{3i\pi}</code>
bold: $3i\pi$	<code>\mathbf{3i\pi}</code>	Ψ	<code>\boldmath\Psi\unboldmath</code>

4.0.4 Other mathematical examples

Here are a few more examples of additional math modes in $\text{L}^\text{A}_\text{T}_\text{E}_\text{X}$:

$$\backslash [\backslash \text{lim}_{\{n\}} \backslash \text{rightrightarrow} \backslash \text{infty} \} x = 1 \backslash] \quad \Longrightarrow \quad \lim_{n \rightarrow \infty} x = 1$$

$$\overbrace{w+\underbrace{x+y}_b+z}^a \implies$$

```

\begin{equation}
x \ll y_{2} + \cdots + y_{n}
\end{equation}

```

$$\implies x \ll y_2 + \cdots + y_n \quad (4.1)$$

```

\[\begin{array}{ccl}
a+b+c & xy & x-y \\
d+e & yz & y-z \\
f & xz & z-x
\end{array}\]

```

$$\implies \begin{array}{ccl}
a+b+c & xy & x-y \\
d+e & yz & y-z \\
f & xz & z-x
\end{array}$$

```

\begin{eqnarray}
x & = & 127y \\
y & > & a+b+\cdots+m \\
& & n+o+p+q
\end{eqnarray}

```

$$\implies \begin{array}{l}
x = 127y \\
y > a + b + \cdots + m + \\
n + o + p + q
\end{array} \quad (4.2)$$

$$(4.3)$$

Use `eqnarray*` instead of `eqnarray` to produce an equation array without equation numbers.

`\usepackage{subeqn}` can also be used in the preamble of your document to control subequation numbering. Then you can then use `\subequations` to start subequation numbering, `\newsubequation` to start a new set of subequations, or `\nosubequations` to turn off subequation numbering; at the start of a subequation line. Subequation numbering can span separate environments and can be turned off in a different environment than where it began. The `\label` and `\ref` commands can be used in subequations. Additionally, the `subeqn` package allows you use the `\eqlabel` command instead of `\label`, which allows you to refer to the whole equation at once. `\eqlabel` can be used anywhere within the subequation.

4.1 Lining up on a decimal point

Since LaTeX uses proportional-pitched fonts with different sized letters/numbers, it is normally difficult to align a group of numbers on a decimal point. One way to do this is to use the typewriter font which is a fixed-pitch font, while another way is to use the exact same number of letters on each side of the decimal point. A more satisfactory way to do this in a tabular environment is to split a number across two columns with a `r@.l` and then use the `\multicolumn` command to span columns in the header. For example:

```

\begin{tabular}{|r@{.}|r@{.}|r@{.}|}
Group & \multicolumn{2}{c}{Reading}
& \multicolumn{2}{c}{Writing} \\
Fifth Grade & 3&32 & 76&52 \\
Sixth Grade & 219&927 & 114&9542 \\
\end{tabular}

```

Group	Reading	Writing
Fifth Grade	3.32	76.52
Sixth Grade	219.927	114.9542

You can also use `@{\extracolsep{\fill}}` as a tabular formatting option to fill up blank areas in fixed width tables (`tabular*`).

4.2 Under-/Over-Brackets and Parentheses

In standard L^AT_EX you can horizontally brace together a standard mathematical expression with `\underbrace` and `\overbrace`, as follows:

```


$$\underbrace{a + \overbrace{b + \cdots + y}^{24} + z}_{26}$$


```

which produces:

$$\underbrace{a + b + \cdots + y + z}_{26}$$

Additionally, there is a L^AT_EX package at Iowa State called `mathbarr` (available from <http://css.ait.iastate.edu/Tex/Local/>) that allows you to horizontally group together a mathematical expression using a bracket or a parenthesis, as well as the standard brace. Just add the `mathbarr` package statement to the preamble of your document:

```

\documentclass{article}
\usepackage{mathbarr}
\begin{document}

```

Once you do that, the following commands are available for your use: `\overbrack`, `\underbrack`, `\overparen` and `\underparen`. The use of the `\overbrack`, `\underbrack`, `\overparen` and `\underparen` are much the same as the `\overbrace` and `\underbrace` commands as shown:

```


$$\overbrack{x+\cdots+x}^{k\rm\;times} \quad \underbrack{x+y+z}_{>\,0}$$


$$\overparen{x+\cdots+x}^{k\rm\;times} \quad \underparen{x+y+z}_{>\,0}$$


```

which produces:

$$\overbrace{x + \cdots + x}^{k \text{ times}} \quad \underbrace{x + y + z}_{>0} \quad \overbrace{x + \cdots + x}^{k \text{ times}} \quad \underbrace{x + y + z}_{>0}$$

The resultant grouping brackets and parentheses look a bit thick but are the exact same thickness as with the standard braces.

4.3 Hypothesis and Bold Lowercase Greek Characters

Two mathematical problems in L^AT_EX are creating hypothesis and bold, lowercase greek characters.

4.3.1 Hypothesis

You can create your own environment in L^AT_EX for stating a hypothesis or theorem. For this type of problem, create a hanging indent style theorem by using the `\newtheorem` command in the preamble of your document:

```
\newtheorem{guess}{Hypothesis}[chapter]
```

Then, for each hypothesis enter:

```
\begin{guess}
\begin{minipage}[t]{4.5 in}
This is my theory which is correct which shows that my theory is correct and
it belongs to me and thus no one else. I composed this theory using two
separate experiments which I also created which are not contained in this
document at all.
\end{minipage}
\end{guess}
```

The output from my above example would look like this:

Hypothesis 4.1 *This is my theory which is correct which shows that my theory is correct and it belongs to me and thus no one else. I composed this theory using two separate experiments which I also created which are not contained in this document at all.*

The `\newtheorem` command allows you to create an environment, which you name (in my example I named it guess), that will create for you a theorem-like environment that may optionally be numbered at a specified

sectional unit level (in my example I numbered it at the chapter level). The `\newtheorem` command declaration should be placed in the preamble (between the `\documentclass` line and the `\begin{document}` line) but can also be placed anywhere in your document before the environment that it creates is used. The `minipage` command used with the `\newtheorem` environment gives the hanging indent-style output that major professors and thesis committee members often look for in a thesis-style paper.

4.3.2 Bold, Lowercase Greek Characters

Bold, lowercase greek characters used to be troublesome to create in L^AT_EX but now the current version of L^AT_EX makes it much easier to get these characters. To get an uppercase greek bold letter, simply use `\mathbf` such as `2ft \Psi \psi` to produce:

2^{ft}Ψψ

This gets you bold uppercase greek letters but not bold lowercase greek letters. To get bold lowercase greek letters, you need to use the `\boldmath` command instead such as `2ft \Psi \psi` `\unboldmath` to produce:

2^{ft}Ψψ

4.4 Putting a Box around a Parbox with Math Formula

Some professional publications encourage you to put a box around a parbox that contains math formula centered on the page like in `displaymath` mode. You can do this using the `\framebox` and `\parbox` commands. Here is a text-only example:

```
This is a line of text.
\framebox{\parbox[t]{1 in}{This is a box aligned along its top
line.}} AND
\framebox{\parbox[b]{1.5 in}{Now a slightly wider box aligned
on its bottom line.}}
```

which produces:

This is a line of text. This is a box aligned along its top line. AND Now a slightly wider box aligned on its bottom line.

While here is something with a centered math equation:

```
\begin{center}
\framebox{\parbox[b]{2 in}{This is a centered box with a math formula
somewhere  $\int_a^b f(x) dx$  inside it.}}
\end{center}
```

which produces:

This is a centered box with
a math formula somewhere
 $\int_a^b f(x) dx$ inside it.

4.5 The Amsmath and Latexsym Packages

If you don't find what you need in standard \LaTeX then you might try using the `amsmath` package that was created for mathematicians by the American Mathematical Society. The `amsmath` package is a \LaTeX version of `amstex` and is described in detail in "The \LaTeX Companion" ([3]). The `amsmath` package gives you some nice additions to standard \LaTeX mathematics also the `latexsym` package gives you some additional mathematical symbols that are not in standard \LaTeX .

Capítulo 5

Accents and Writing in Español with the Babel Package

5.1 General L^AT_EX Accents and Babel

L^AT_EX by default contains many accents and Non-English symbols:

<code>\' {a}</code> ⇒ á	<code>\~ {a}</code> ⇒ ã	<code>\v {a}</code> ⇒ ǎ	<code>\c {a}</code> ⇒ ç
<code>\' {a}</code> ⇒ à	<code>\= {a}</code> ⇒ ā	<code>\H {a}</code> ⇒ ħ	<code>\d {a}</code> ⇒ đ
<code>\^ {a}</code> ⇒ â	<code>\. {a}</code> ⇒ ȧ	<code>\t {aa}</code> ⇒ ââ	<code>\b {a}</code> ⇒ ß
<code>\" {a}</code> ⇒ ä	<code>\u {a}</code> ⇒ ů	<code>\oe</code> ⇒ œ	<code>\aa</code> ⇒ å
<code>\l</code> ⇒ l	<code>? ' ⇒ ĵ</code>	<code>\OE</code> ⇒ Œ	<code>\AA</code> ⇒ Å
<code>\L</code> ⇒ L	<code>! ' ⇒ j</code>	<code>\ae</code> ⇒ æ	<code>\o</code> ⇒ ø
<code>\ss</code> ⇒ ß	<code>\AE</code> ⇒ Æ	<code>\O</code> ⇒ Ø	<code>\dag</code> ⇒ †
<code>\S</code> ⇒ §	<code>\copyright</code> ⇒ ©	<code>\ddag</code> ⇒ ‡	<code>\P</code> ⇒ ¶
<code>\pounds</code> ⇒ £	<code>\i</code> ⇒ ı	<code>\j</code> ⇒ j	<code>\= {\i}</code> ⇒ ī

However, if you are going to be writing a long passage or an entire document in a language other than English then standard L^AT_EX can be cumbersome. There is a package called babel which allows you to make another language “standard” in an English L^AT_EX document. An example of spanish Babel follows but the current full version of L^AT_EX contains language files for: Afrikaans, Bahasa, Basque, Breton, Bulgarian, Catalan, Croatian, Czech, Danish, Dutch, English, Esperanto, Estonian, Finnish, French, Galician, German, Greek, Hebrew, Hungarian, Icelandic, Interlingua, Irish Gaelic, Italian, Latin, Lower Sorbian, North Sami, Norwegian, Polish, Portuguese, Romanian, Russian, Scottish Gaelic, Spanish, Slovakian, Slovenian, Swedish, Serbian, Turkish, Ukrainian, Upper Sorbian and Welsh... and others.

To load a babel package, use:

```
\usepackage[spanish,english]{babel}
```

If you list more than one babel option, the last option that you choose will be active when your document begins. You can switch to another language, like I did for this chapter, by using the `\selectlanguage` command:

```
\selectlanguage{spanish}
```

General babel package information and some specific language information can be found in “The L^AT_EX Companion” ([3]). Specific package information can be found at CTAN (the Comprehensive T_EX Archive Network: <http://www.ctan.org/tex-archive/macros/latex/required/babel/>).

5.2 Using Spanish Babel: An Extended Example

Hola! The easiest way to create a document in Spanish with L^AT_EX is to use the spanish option within the Babel package:

```
\usepackage[spanish]{babel}
```

which is normally placed in the preamble to your document.

The spanish babel package automatically changes all the heading and titling information that L^AT_EX creates for your document into Spanish. For instance, L^AT_EX changes figures from ‘Figure’ to ‘Figura’, a ‘Table’ gets changed to a ‘Tabla’ and the ‘Table of Contents’ becomes ‘Índice General’.

The spanish babel package also includes some shorthand options for writing in Spanish as well as also including the standard Spanish symbols available in normal L^AT_EX. Table 5.1 shows a summary of all Spanish symbols available in the L^AT_EX spanish babel package.

Here is an example of a paragraph written with spanish babel:

Ésta es Gabriela Sabatini, de la Argentina. Yo también juego al tenis, pero no soy bueno y famoso como ella. Sólo comencé a jugar al tenis el año pasado. Compré una raqueta y tres pelotas y practiqué con un experto en una cancha de la universidad. Este año no gané varios partidos, pero perdí el campeonato. ¡Qué mala suerte!¹

which was produced from the following lines of L^AT_EX code:

¹*¡El Camino! A Cultural Approach to Beginning Spanish* (fourth edition) by Nicholas, Swietlicki, Dominicus & Neale-Silva, 1990, McGraw-Hill, page 157.

Cuadro 5.1: Spanish Characters Available in the L^AT_EX Spanish Babel Package

?	?	?’	¿
!	!	!‘	¡
"u	ü	"U	Ü
"<	«	">	»
"o	.º	"a	.ª
\’a	á	\’A	Á
\’e	é	\’E	É
\’{\i}	í	\’I	Í
\’o	ó	\’O	Ó
\’u	ú	\’U	Ú
\’n	ñ	\’N	Ñ
~n	n	~N	N
\"-	¨	\"	¨
\’,	´	\’‘	´

\spanishhyphenmins ΘΘ \today 28 de enero de 2011

\’Esta es Gabriela Sabatini, de la Argentina.
 Yo tambi\’en juego al tenis, pero no soy bueno y
 famoso como ella.
 S\’olo comenc\’e a jugar al tenis el a~no pasado.
 Compr\’e una raqueta y tres pelotas y practiqu\’e
 con un experto en una cancha de la universidad.
 Este a~no gan\’e varios partidos, pero perd\’{\i} el
 campeonato. !‘Qu\’e mala suerte!

A complete spanish babel example spanishbabel.tex is available at the
 Iowa State Tex web site: <http://css.ait.iastate.edu/Tex/>.

Adios!

Chapter 6

Creating a Presentation with L^AT_EX

One of the key advantages to L^AT_EX is that you can use L^AT_EX to create mathematical formulas and information. If you need to do this for a presentational handout or a technical presentation, there are a couple of options that you can choose from with L^AT_EX in order to preserve your mathematical display and make the transition from paper to presentation.

6.1 The Slides Documentclass

A traditional way to create a presentation with L^AT_EX is to use the slides document class. Within the slides document file, you would start your document as normal for a standard-style overhead set of slides: `\documentclass{slides}`; or as follows for a more presentational, landscape set of slides:

```
\documentclass[landscape]{slides}
\usepackage[landscape]{geometry}
```

Then each page or slide in your handout would contain the slide environment:

```
\begin{slide}
  GNATS and GNUS
\end{slide}
```

The slides documentclass fonts are made much larger than a traditional L^AT_EX document and L^AT_EX commands that don't make sense within a slides-

like area such as the table and figure environments, sectional commands and page-breaking commands; are not allowed.

However, you can setup your own set of sectional-like commands in the preamble of the document:

```
\documentclass{slides}
\usepackage[pdftex]{graphicx}
\usepackage{fancybox}
%\usepackage{color}
\newcommand{\topic}[1]{%
  \noindent
  \begin{center}
    \bfseries
    \shadowbox{#1}%
  \end{center}}
\newcommand{\subtopic}[1]{%
  \noindent
  \ovalbox{
    \begin{minipage}{\textwidth}
      \bfseries
      #1 %
    \end{minipage}}}

```

So using this method an individual slide would be something like:

```
\begin{slide}
\topic{General Flash Information}

\vspace*{-.25 in}
Adobe Flash is a software authoring tool
for the Web.

\subtopic{Starting Flash}

\vspace*{-.25 in}
\small
To begin Flash on a Macintosh, have the Finder active and go under

```

The slides documentclass can use a standard title page and most other standard L^AT_EX commands. More information on the slides documentclass can be found in any standard L^AT_EX manual or online you might try going to: <http://texpblog.wordpress.com/2008/01/21/create-your-slides-presentations-with-latex/>

6.2 The Beamer Documentclass

The beamer documentclass is a new approach to doing more of a Powerpoint-like presentation with L^AT_EX. You start off a Beamer document by declaring the documentclass then choosing a theme for your presentation and giving title page information:

```
\documentclass{beamer}
\usetheme{default}
\title{A Rift in Time}
\author{Joe Struss}
\institute{Lakeland Shores}
\begin{document}
```

There are many themes to choose from named after the various cities where the Beamer's author and collaborators have made Beamer presentations like: Boadilla, Warsaw, Madrid, Pittsburgh, Rochester, Copenhagen, Malmoe and Singapore. Other options include color themes, inner themes and outer themes. A good Beamer Quickstart Guide showing a lot of the themes and options can be found at: <http://www.math.umbc.edu/~rouben/beamer/>.

Then each slide in your Beamer document is a frame-environment within L^AT_EX with the added ability to give each frame a title or to use your title information by using the standard `\titlepage` command:

```
\begin{frame}
\titlepage
\end{frame}

\begin{frame}{Introduction}
This is a short introduction.
\end{frame}
```

One of the nice features of Beamer is that you can delay parts of the page appearing on the screen with the Animation overlay options that come with Beamer like the `\pause` and `<n- >` commands:

```
\begin{frame}{Following on}
\begin{itemize}
\pause \item Hello
\pause \item Jello
\pause \item Mellow
\end{itemize}
\end{frame}
```

```
\end{itemize}
\end{frame}
```

```
\begin{frame}{Continuing on}
\begin{itemize}
\item<2-3> appears on 2nd and 3rd
\item<3> appears on 3rd only
\item<4-> appears from 4th on
\item<5-> appears from 5th on
\end{itemize}
\end{frame}
```

Beamer is a terrific tool for creating a Powerpoint-like presentation with \LaTeX but does take a little time to learn the package and all the options you have to choose from within Beamer. A quick introduction to Beamer can be found here: <http://www.math-linux.com/spip.php?article77>

While complete Beamer information can be found at CTAN:
<http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf>.

6.3 3-D Objects

Another new \LaTeX development is the option to add 3-D objects within a PDF. MeshLab and U3D can be used for doing this sort of thing and can add a whole new dimension to a paper or a presentation.

Complete MeshLab information with links to U3D document information can be found at: <http://meshlab.sourceforge.net/>

An example of using MeshLab/U3D within a \LaTeX document can be found at:

<http://meshlab.sourceforge.net/wiki/images/c/cc/Laurana.pdf>

Quite remarkable.

Appendix A

Font Types and Styles

There are four characteristics that can describe any font: shape, series, family and size. L^AT_EX allows you to individually change each of these characteristics for any textual item within a document. The type styles (shape, series and family) can also be invoked in either a “closed form” as a command `\cmd{the}` or in an “open form” as a declaration `{\decl the}`. The “closed form” is useful for changing a word or two of text while the “open form” when used without a brace is useful for changing large sections of text. Here are some examples from each of the four groups (*- default):

Shape Styles:	Closed Form	Open Form	
Upright shape*	<code>\textup{the}</code>	<code>{\upshape the}</code>	
Italic shape	<code>\textit{the}</code>	<code>{\itshape the}</code>	
Slanted shape	<code>\textsl{the}</code>	<code>{\slshape the}</code>	
Small Caps shape	<code>\textsc{the}</code>	<code>{\scshape the}</code>	
Series Styles:	Closed Form	Open Form	
Medium series*	<code>\textmd{the}</code>	<code>{\mdseries the}</code>	
Boldface series	<code>\textbf{the}</code>	<code>{\bfseries the}</code>	
Family Styles	Closed Form	Open Form	
Roman family*	<code>\textrm{the}</code>	<code>{\rmfamily the}</code>	
Sans serif family	<code>\textsf{the}</code>	<code>{\sffamily the}</code>	
Typewriter family	<code>\texttt{the}</code>	<code>{\ttfamily the}</code>	
Sizes:	Open Form Only	(listed from smallest to largest)	
<code>{\tiny a}</code>	<code>{\small a}</code>	<code>{\large a}</code>	<code>{\huge a}</code>
<code>{\scriptsize a}</code>	<code>{\normalsize a}</code>	<code>{\Large a}</code>	<code>{\Huge a}</code>
<code>{\footnotesize a}</code>		<code>{\LARGE a}</code>	

The normalfont command or declaration: `\textnormal{the}` or `{\normalfont the}` returns the text to the default font for that documentclass.

Appendix B

PDF Files and .EPS Graphics

Previous versions of \LaTeX (and current users of \LaTeX who want to use the standard \LaTeX compiler) can only use .ps and .eps graphics; but those graphic types cannot be put directly into a PDF document for use with $\text{PDF}\text{\LaTeX}$ which is the most popular current \LaTeX compiler. To get around this— if you have a .ps or .eps graphic that you would like to use in your document- try using the **epstopdf** package immediately after the graphicx package in your document preamble. The epstopdf package will try and convert your .ps or .eps graphic into a pdf before including it in your document. Just after your document starts, you can also use the command:

```
\DeclareGraphicsExtensions{.jpg,.pdf,.mps,.png}
```

which declares a suffix for your graphic files so you don't have to declare it with every `\includegraphics` command:

```
\includegraphics{nook2c}
```

By default, the normal graphicx package without the pdftex option will be looking for files ending in .ps if a suffix isn't given. So this allows for increased compatibility for switching between the \LaTeX and $\text{PDF}\text{\LaTeX}$ compilers if you just leave off the suffix on your graphic files.

Appendix C

General L^AT_EX Information

Local Iowa State information on L^AT_EX can be found at:

<http://css.ait.iastate.edu/TeX/>

included in this web site are links to the isuthesis package, the isuthesis documentclass files, the isuthesis1file package and the isuthesis template files; amongst other local sundries.

You can download a full version of L^AT_EX to use on most standard microcomputers currently in use from the T_EX User's Group. The TeX User's Group is the international home for use of L^AT_EX and T_EX throughout the globe. This central source for all wisdom and knowledge that is L^AT_EX and TeX related can be found at:

<http://www.tug.org/>

Appendix D

Basic L^AT_EX Commands on Project Vincent or a UNIX system

To use L^AT_EX on Project Vincent or a standard UNIX system, you will need to use the following set of commands:

% **add web2c** — needs to be entered once during a session to access the web2c locker and make L^AT_EX commands available for your use (Project Vincent only).

% **pdflatex filename.tex** — compiles a L^AT_EX file, checks for correct syntax and if there are no errors produces a filename.**pdf** file as a result. The pdflatex command requires all graphic files to be .pdf or .jpg in format instead of the standard L^AT_EX postscript (.ps) graphic file.

If you get a ? then you have an error in your file. Enter **X** to exit the compiler, correct your error and then recompile your filename.tex all over again. You can also enter a ? to get other options.

% **latex filename.tex** — compiles a L^AT_EX file, checks for correct syntax and if there are no errors produces a filename.**dvi** file.

If you get a ? then you have an error in your file. Enter **X** to exit the compiler, correct your error and then recompile your filename.tex all over again. You can also enter a ? to get other options.

% **add ghost** — needs to be entered once during a session to access the ghost locker and make Ghostscript commands available for your use (Project Vincent only).

- % **xdvi filename.dvi** — allows you to view the .dvi output from your \LaTeX file.
The xdvi normally will not display document graphics. You must use **add ghost** to add the ghost locker and thus give Ghostscript capabilities to xdvi in order to show postscript graphic files within xdvi. This command will only work in an X-Windows environment like a Project Vincent workstation.
- % **dvips filename.dvi** — creates a filename.ps file out of your filename.dvi file which you can then print with the **lpr** command.
- % **gs filename.ps** — allows you to view the .ps output including graphics from your \LaTeX file. This command will only work in an X-Windows environment like a Project Vincent workstation.
- % **ispell -t filename.tex** — will spell check a \LaTeX document and ignore \LaTeX commands and symbols. Once you are inside of ispell enter a ? for additional commands and information.

Here is a sample set of commands that are used to compile and display the latex file named **one.tex**. Only enter what is listed in bold; for example, **latex one.tex**— as the % is simply the UNIX prompt.

- % **add web2c** - adds the web2c locker (needs to be done once per session).
- % **pdflatex one.tex** - compiles one.tex with \LaTeX and produces a pdf file named one.pdf.
- % **latex one.tex** - compiles one.tex with \LaTeX and produces a dvi file named one.dvi.
- % **add ghost** - adds the Ghostscript locker (needs to be done once per session).
- % **xdvi one.dvi** - views the one.dvi file with the xdvi previewer.
- % **dvips one.dvi** - creates a one.ps for use in printing.
- % **lpr one.ps** - prints the file named one.ps.
- % **ispell -t one.tex** - checkes one.tex for spelling errors using the -t option in ispell.

If you are using \LaTeX on a Windows or Macintosh, you would normally compile your .tex document with the PDF \LaTeX compiler by pressing a Build or Compile button and then simply view the resulting .pdf file. See [4] for more information

References

- [1] *L^AT_EX User's Guide and Reference Manual* by Leslie Lamport; Standard L^AT_EX manual with lots of useful information. A little expensive because it is the “official” L^AT_EX book.
- [2] *A Guide to L^AT_EX* by Helmut Kopka and Patrick W. Daly; A great, less expensive L^AT_EX book that covers both standard L^AT_EX and gets into some L^AT_EX information regarding graphics.
- [3] *The L^AT_EX Companion* by Goosens, Mittelbach and Samarin; The “official” follow-up guide to Lamport's book. Terrific information on L^AT_EX packages but still a little pricey.
- [4] *Installing T_EX/L^AT_EX on a Windows or Macintosh System* by Joe Struss; A step-by-step guide to installing L^AT_EX on a Windows or Macintosh system including information on running an initial L^AT_EX file and adding the local isuthesis package to your system. Available as a free download from: <http://css.ait.iastate.edu/Tex/classes.html>
- [5] *The isuthesis Package* by Joe Struss; More information than you would ever want on the isuthesis package, the isuthesis1file package and the isuthesis document class; any of which can be used to produce a thesis at Iowa State University. Available as a free download from: <http://css.ait.iastate.edu/Tex/classes.html>