

Systems Lab L^AT_EX Template Commentary

Section #, Group #:
Group Member 1¹ Group Member 2¹ Group Member 3¹ Group Member 4¹
GTA: Best GTA Ever¹

Prepared By:
Michael Rouleau²

18th February 2019

¹Missouri University of Science and Technology, Rolla, Missouri 65409

²Georgia Institute of Technology, Atlanta, Georgia 80813

Chapter 1

Using L^AT_EX

1.1 About L^AT_EX

L^AT_EX, pronounced "Lah-tech" or "Lay-tech" [2], is an advanced document preparation system popular in the scientific world and academia. Instead of organizing a document visually like a WYSIWYG (what you see is what you get) package (e.g. Microsoft Word), L^AT_EX has a logical structure that focuses on content rather than formatting decisions. This means that if you decide that all of your fractions of the form $\frac{x}{y}$ require a little extra size, as in $\frac{x}{y}$, the entire document can be reformatted with just a line or two of code. Similarly, it is possible to compile a list of sources that were used during research, then automatically list and number only those which were cited in the manuscript. Citations auto-update. Updating figures is a breeze. Complex mathematical notations are just a few keystrokes away. For long scientific documents, L^AT_EX is the cat's meow.

1.2 Installation

To start editing a L^AT_EX document on a Missouri S&T Computer, simply log on to AppsAnywhere and start the program "Texmaker."

For personal computers, Texmaker is freely available from <http://www.xmlmath.net/texmaker/>; its dependency, MiKTeX, is available from <https://miktex.org/>. Installing MiKTeX before Texmaker is highly recommended. Detailed installation instructions are available on each package's respective website. It is possible to run a portable installation of the above from a USB drive.

Online editors are also available - Overleaf [3], for example, is a free collaborative editor somewhat similar to Google Docs.

1.3 Basics

1.3.1 Syntax

L^AT_EX code is typed into a plain-text file called an "input" or "source" file that contains markup which tells the compiler how to display content. In general, the following reserved characters demarcate normal text from commands or other special functions:

\$ % ^ & _ { } ~ \.

It is still possible to use these symbols, but they typically require a backslash prefix or other special notation.

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

Commands also usually require a backslash prefix. For example, the command `\textbf{text}` turns **text** bold. Most commands can be nested; `\textbf{\textit{Italics and Bold!}}` turns text both ***Italics and Bold!***

Likewise, backslashes are required to begin an environment. Environments act similar to commands but are usually applied to multiple lines or used for more complex operations. For example, the following code would insert a picture file named “joeminer” located in the directory where the L^AT_EX file is saved.

```
1 \begin{figure}
2   \includegraphics{joeminer}
3 \end{figure}
```

Many commands and environments are packaged with standard L^AT_EX distributions, but more customization is available by loading external packages. One of the most popular packages, `graphicx`, allows users to load and manipulate figures, as we did with the joeminer figure above.

Coding comments can be added with the `%` symbol. Following a percent symbol, the compiler will ignore the remainder of the current line and any whitespace at the beginning of the following line. This can be useful for adding notes that should not be included in the final document.

1.4 Structure

The minimum structure of a L^AT_EX document is best illustrated with the following Hello World example. Feel free to follow along in Texmaker.

```
1 \documentclass{article}
2 \begin{document}
3   Hello World!
4 \end{document}
```

Breaking down each line, we first have an initial declaration of which class the document is - this sets basic definitions for formatting. In this example, we’re using the article class, which is a common starting point.

The second line is where the document officially starts. This is where the environment called “document” begins - all of your content will go between the `\begin{document}` and `\end{document}` tags. Everything before this point is traditionally called the “Preamble.” If you are including any external packages, changing global formatting, or redefining commands, do so in the Preamble.

Our content, of course, only consists of “Hello World!”. This is where your entire report, book, manuscript, etc. would go.

Finally, `\end{document}` ends the document environment. Anything after this line is ignored.

To compile your first L^AT_EX document, save your file (without spaces in the name!) and click the “Run” arrow in the top toolbar of Texmaker. Your new document should appear on the right side of the window. For simple documents, only one compilation is necessary. For complex documents (especially those with intricate cross-references), multiple compilation steps may be required.

1.5 Organization

\LaTeX commands for breaking up documents are intuitive. For example, paragraphs are created by placing a blank line in your code before the start of the next paragraph. Any more than 1 blank line is ignored. The command `\chapter{Silver}` starts a new chapter named “Silver” and `\section{Gold}` starts a new section called “Gold.” It is left as an exercise for the reader to discover what `\subsection{text}` and `\subsubsection{text}` do.

With each of these commands, a number or other counter is inserted by default. To create an unnumbered heading, the “starred” version may be used. For example, a unnumbered section heading would use the command `\section*{text}`. To manually adjust a counter, the command `\setcounter{countername}{#}` is used.

In \LaTeX , we can also define containers called “floats.” These containers are traditionally applied to things that cannot be broken over a page, like figures and tables; these are floats by default. To keep these items intact, \LaTeX moves floats around to a spot where they can be preserved. Since by definition these containers move around, it is beneficial to define a list of preferred locations to place the container. To do this, each float declaration takes a position argument. A handful of these options are listed below in Table 1.1 [3]. Note that too many floats with the H specifier in a row may cause errors.

Table 1.1: Float Position Options

Option	Permission
h	Place the float here, i.e., approximately at the same point it occurs in the source text.
t	Position at the top of the page.
b	Position at the bottom of the page.
p	Put on a special page for floats only.
!	Override internal parameters <code>\LaTeX{}</code> uses for determining “good” float positions.
H	Places the float at precisely the location in the <code>\LaTeX{}</code> code.

Figures, tables, etc. may also be assigned a label using `\label{key}`. This label isn’t displayed in the compiled document, but allows for automatic cross-referencing elsewhere in the document. Referencing is covered in Section 2.3.

Example

Let's modify our Hello World code to make use of several of the above features. Before compiling, make sure there is an image named "figure1" in the same directory as your code. Most common file types are supported (jpg, png, etc).

```
1 \documentclass{article}
2 \usepackage{graphicx}
3 \begin{document}
4     \section{A Clever Heading}
5         \label{sect:Clever}
6         This section has a number and a label!
7     \section*{Figures}
8         This section does not have a number displayed, but it
9         has a figure on a dedicated page for floats!
10        \begin{figure}[p]
11            \includegraphics{cat-painting}
12        \end{figure}
13 \end{document}
```

Chapter 2

The Template

2.1 Preamble

With these basics under our belt, we can decode the Lab Template. Let's look at the preamble together.

```
1 %Template and commentary by Michael Rouleau: https://github.com/
   mikerouleau/ME4842.git
2 %Text primarily from Dr. Stutts' Systems Laboratory Manual
3 %Last Updated: Feb 17, 2019
4
5 %——This template includes equation captions and numbering BELOW the
   equation——%
6
7 \documentclass[11pt,letter]{report}
8 \usepackage[margin=1in]{geometry}
9 \usepackage[justification=centering,format=plain,font=it,margin=1
   in]{caption}
10 \usepackage{amsmath,amssymb}
11 \usepackage{graphicx}
12 \usepackage{float}
13 \usepackage{aliascnt}
14
15 \newaliascnt{eqfloat}{equation}
16 \newfloat{eqfloat}[!htb]{eqflts}
17 \floatname{eqfloat}{Equation}
18
19 \title{\Huge{Systems Lab \LaTeX\ Template Commentary}}
20 \author{Section \#, Group \#: \\
21         Group Member 1 \footnote{Missouri University of
22         Science and Technology, Rolla, Missouri 65409}
23         \quad Group Member 2\footnotemark[1]
24         \quad Group Member 3\footnotemark[1]
25         \quad Group Member 4\footnotemark[1]
26         \\ GTA: Blaine Allen \\ \\ }
```

```

26      Prepared By: \\ Michael Rouleau\footnote{Georgia
27      Institute of Technology , Atlanta , Georgia 80813}
28  }
29  \date{\today}

```

Lines 1-6 contain notes to whoever is reading the code. It is good practice to comment your name and date as a bare minimum, especially if someone may look at your source code later.

Line 7 defines the document class, just like we did in Section 1.4. The only difference is now we have also specified values for a couple of options: paper size and normal text font size. While these may be set by a class, using an option overrides the default value.

Lines 8-13 all include various external packages for increased customization. The geometry package allows us to define margins of 1". Caption changes the default font settings for our captions. Amsmath and amssymb provide some fancy commands and environments for math (note we're being sneaky and including two packages at once on this line). We'll primarily be using the equation environment. Graphicx allows us to insert figures with `\includegraphics{imagefile}`. Finally, the float and aliascnt packages are used for some fancy manipulation of equations in order to meet the formatting requirements.

Speaking of formatting gymnastics, lines 15-17 are where the most complex of it takes place. In line 15, we define a new alias count called "eqfloat." This alias essentially acts as another name for the counter it follows, "equation." This means if eqfloat is indexed, equation will be indexed and vice versa. Line 16 takes eqfloat and defines a float container also named "eqfloat". [lhtb] defines our preferred placements, and "eqflts" is the name it uses for the log file. The float implicitly inherits the counter alias from line 16 - each eqnfloat environment is counted with the equation counter. Line 17 changes the name of the float to be "Equation." By doing this, the `\caption{A Clever Caption}` command will read *Equation #: A Clever Caption*, instead of *eqfloat #: A Clever Caption*.

Finally lines 19-28 pertain to the title page. This section is fairly self-explanatory. However, you'll note the use of `\quad` and `\\` within the author command. `\quad` helps with the spacing of authors. Even if you don't have 4 group members, `\quad` will likely still work to distribute them in an aesthetic manner. `\\`, on the other hand, serves as a complex macro that introduces a line break in this context. Limit the use of `\\` unless necessary. Without adding additional packages, one way to add author affiliations is through the use of footnotes. The first author with a given affiliation gets the `\footnote{My School}` command. Each following author gets `\footnotemark[#]` to designate which affiliation they belong to. With this implementation, the affiliation number using `\footnotemark[#]` does NOT auto-update. Be careful here. If you wish to explore a more flexible option, take a look at the authblk package.

2.2 Body

After the Preamble, `\begin{document}` begins the document environment. Immediately after, we use the `\maketitle` command to generate the title page of our document. Since the document body usage is repetitive, let's break up the template further into different environments and their usage.

2.2.1 Sections

Just as described in Section 1.5, the command `\section*{Section Title}` defines the beginning of a new section. The “starred” version of this command removes the section number from showing up on your final report. For each new section, simply add the `\section{}` command and type the name of the new section in curly brackets following it. Subsections may be nested within sections.

2.2.2 Tables

Tables are perhaps the most difficult format to insert in L^AT_EX. Fortunately, we can logically break down each line again.

```
1      \begin{table} [!htb]
2          \caption{Annual Per Capita Consumption of Mozzarella
3              Cheese and Civil Engineering Doctorates Awarded \
4              cite{Vigen}}
5          \label{table:Civil Cheese}
6          \centering
7          \begin{tabular} {|c|c|c|}
8              \hline
9              \textbf{Year} &
10             \begin{tabular} {c} {@} {c} {@} {}
11             \end{tabular} &
12             \begin{tabular} {c} {@} {c} {@} {}
13             \end{tabular} \\
14             Civil Engineering \\\
15             Doctorates
16             \end{tabular} &
17             \begin{tabular} {c} {@} {c} {@} {}
18             \end{tabular} \\
19             Mozzarella \\\
20             Cheese Consumption
21             \end{tabular} & \\\ \hline
22             2000 & 480 & 9.3 & \\\ \hline
23             2001 & 501 & 9.7 & \\\ \hline
24             2002 & 540 & 9.7 & \\\ \hline
25             2003 & 552 & 9.7 & \\\ \hline
26             2004 & 547 & 9.9 & \\\ \hline
27         \end{tabular}
28     \end{table}
```

At this point, you should be able to guess what line 1 does - it marks the beginning of a table environment. Since tables are floating environments, the `[!htb]` tag tells the compiler where to place the table, in order of preference. Refer to Section 1.5 for more options. Line 2 adds a caption for the table, complete with a citation - we’ll get to those later, in Section 2.3. With this template, each table is automatically numbered as long as a caption exists. Line 3 adds a reference label to the table. Again, we’ll cover what this means in Section 2.3. Line 4 tells the compiler to center the current paragraph, or in this case, table.

Line 5 is where the fun begins. First, we start a tabular environment within the table environment and define the number of columns (3) using `{|c|c|c|}`. Note that `c` does not refer to “column.” Instead, `c` refers to centered - the formatting for each column’s data. Other popular options are “l” for left and “r” for right alignment. Line 6 inserts a horizontal line that is the width of our table.

Lines 7-15 start off easy - the first cell in our table, moving left to right from the top is Year, in bold text. `&` signifies the end of the cell, and we move into our second cell on line 8. Picking

this cell apart, we have a `\textbf{}` command again - so this cell is also in bold text. Within it, another tabular environment starts. This second environment allows us to create subcells within the cell - one for “Civil Engineering” and the other for “Doctorates.” Unfortunately, the standard tabular package does not support `\newline` or `\linebreak`, so this is the best native workaround I am aware of. In declaring this new tabular environment, we use `[c]{@{}c@{}}` - different from the first time. `[c]` in this case is our vertical alignment - again, centered. Other options include `t` for top and `b` for bottom. Our horizontal parameters are taken care of within the curly brackets. `@{}` overrides any horizontal padding on each side of these subcells, and `c` is again horizontal alignment for the single column of subcells. The contents of each subcell is on lines 9 and 10, with a `\\` to demarcate the end of a row. With an `$` to move us into the next cell, the same process is repeated for our final title cell. After all three cells in the top row are full, `\\ \hline` is used to mark the end of the row and insert a horizontal line.

Armed with this information, lines 16-22 should be easy to figure out. However, if you find tables confusing, you’re in luck! For small data sets, there are several online tools to automatically generate L^AT_EX tables. One such tool is located at <https://www.tablesgenerator.com/>. Some L^AT_EX editors also include table-building functionality.

2.2.3 Figures

Figure insertion is a comparatively trivial process using the `graphicx` package. Simply by looking at the code below, most of it should be easily understood.

```
1 \begin{figure}[!htb]
2     \label{fig:Cat Lord}
3     \centering
4     \includegraphics[scale=0.4]{cat-painting}
5     \caption{Robert Cheseman (Holbein)\cite{Herbert}}
6 \end{figure}
```

As expected, `\begin{figure}[!htb]` starts a figure environment. Figures are again floating environments, so we provide them with placement preferences. And as we’ve seen before, `\label{key}` defines a label and `\centering` centers the figure.

The `includegraphics` command itself takes the form of `\includegraphics[options]{imagefile}`, where options are things like height, width, and scale. If you use an option such as height or width, be sure to include a unit (e.g. `height=3cm`). Our feline friend is a bit on the large side, so we’ve scaled him down to 40% of his original size. Purrfect! The `imagefile` field takes the filepath of the image as input - no need to add an extension, the compiler will search for all supported formats. Each time the compiler runs, it searches for the figure file and re-inserts it. This means that if you need to swap out a figure, simply upload the new figure to the same directory and rename the new and old files appropriately. Next time the compiler is run, the new figure will appear.

Finally, we have a caption, complete with citation. With this template, each figure is automatically numbered as long as a caption exists - no need to keep track of figure numbers! For more information on citations, refer to Section 2.3.

2.2.4 Lists

A list is a convenient way to format the procedure section of a lab report. For most situations, a numbered list will suffice, taking the form of

```
1 \begin{enumerate}
2     \item First item
3     \item Second item
4     \item Third item
5 \end{enumerate}
```

Here, we begin an `enumerate` environment and indicate the start of each line item as an `\item`. The same process works for an unordered list using the `itemize` environment. The syntax is identical.

For more complex lists, it is possible to nest lists. For example,

```
1 \begin{enumerate}
2     \item First item
3     \item \begin{enumerate}
4         \item First subitem
5         \item Second subitem
6     \end{enumerate}
7     \item \begin{itemize}
8         \item First subitem
9         \item Second subitem
10    \end{itemize}
11 \end{enumerate}
```

generates the following list:

1. First item
2. (a) First subitem
 (b) Second subitem
3. • First subitem
 • Second subitem

2.2.5 Equations

Equations are one of the significant strengths of any \LaTeX distribution. Instead of digging through symbol menus, complex functions can be typed quickly and intuitively.

For short documents that require few equations, displaying equations inline with text may be appropriate. To enter math mode for this usage, enclose symbols in $\$$ signs (e.g. $\$equation\ here\$$).

Alternatively, equations can be displayed separate from the main body of text by using the `eqfloat` and `equation` environments as shown in the following example. Note that these environments are nested to provide appropriate formatting with a label beneath the equation. The equation itself lies within the `equation` environment. As before, the caption is specified with `\caption{}`. A label is added to conveniently reference the equation elsewhere in the main text. When in math mode, feel free to space equations out for code readability - almost all white space is ignored by the compiler.

```

1 \begin{eqfloat}
2   \caption{The Quadratic Formula}
3   \label{eq:quadratic}
4   \begin{equation*}
5     x_1 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
6   \end{equation*}
7 \end{eqfloat}

```

The resulting equation appears as:

$$x_1 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Equation 2.1: The Quadratic Formula

To label the equation using a right-justified number in parentheses, use the non-starred version of a `\begin{equation}` environment. For instance, take the following code and its resulting form.

```

1 \begin{equation}
2   \label{eq:Dirac}
3   \left (
4     \beta mc^2 + c
5     \left (
6       \sum_{n=1}^3 \alpha_n p_n
7     \right )
8     \right )
9     \psi(x, t)
10    =
11    i \hbar \frac{\partial \psi(x, t)}{\partial t}
12 \end{equation}

```

$$\left(\beta mc^2 + c \left(\sum_{n=1}^3 \alpha_n p_n \right) \right) \psi(x, t) = i\hbar \frac{\partial \psi(x, t)}{\partial t} \quad (2.2)$$

2.3 References

References can be hard to keep track of, especially with inline citations. \LaTeX helps keep track of your inline citations by using a reference key or label to specify a certain author, figure, or even section of the same document. As the document changes (e.g. you add a figure earlier in your report), the counter for the respective object changes (remember the counter we aliased for equations?). When this number changes, the number at each citation updates as well, keeping the entire document consistent. For example, take a figure given by

```

1 \begin{figure} [!htb]
2   \label{fig:Cat Lord}
3   \centering
4   \includegraphics[scale=0.4]{cat-painting}
5   \caption{Robert Cheseman (Holbein) \cite{Herbert}}

```

6 `\end{figure}`

Here, we’ve given Sir Cheseaman the label “Cat Lord.” If I want to cite this figure elsewhere in my code, I would type “`Figure \ref{fig:Cat Lord}`” to refer to the painting, regardless of where it is in the document. For this figure, we also use the command `\cite{key}`. Citations refer to the bibliography entries, which might look something like this for a pure L^AT_EX document:

```
1 \begin{thebibliography}{9}
2
3     \bibitem{Lamport}
4     Lamport, L., \emph{LATEX: a document preparation system:
5         users guide and reference manual.} Addison–wesley. 1994.
6
7     \bibitem{Stutts}
8     Stutts, D.S., \emph{Mechanical Engineering Systems Laboratory
9         Manual.}
10    Department of Mechanical and Aerospace Engineering, Missouri
11    University of Science and Technology, 3 Oct. 2018.
12    130–131.
13
14    \bibitem{Vigen}
15    Vigen, T., \emph{15 Insane Things That Correlate With Each
16        Other.}
17    Spurious Correlations, tylervigen.com/spurious-correlations.
18    Accessed 8 Nov. 2018.
19
20 \end{thebibliography}
```

The format here should look similar to a list, and with good reason - what is a bibliography but a list? On line one, we begin the bibliography environment and specify the number of references that are listed. For less than 10 references, the value 9 is commonly used, even if there are only 3 entries as in this case. For 10-99 entries, the value 99 is used.

Each entry starts similar to the items in a list, but we use the `\bibitem{key}` command instead of `\item`. The key argument is analogous to the label we use for equations and figures - this is how you cite the work elsewhere in the document. Just like using `\ref{label}`, we use `\cite{Stutts}` to cite Dr Stutts’ bibliography number as `\cite{Stutts}`.

Complex projects can make use of an automatically generated bibliography, much like our front page. However, the methods to implement such a bibliography lies outside the scope of this document. I encourage interested readers to look up BibTeX reference management tools.

2.4 Including Code

For some experiments, it may be useful to include MATLAB code, especially in the appendix. While this template was not originally designed with that functionality, it can be easily modified to do so.

The first step is to include the `listings` package in the Preamble along with a couple options. To add a colored background for the code, also include `xcolor`. Either way, brace yourself for all the options. It’s easiest to just copy-paste the following code into your Preamble.

```

1 \usepackage{xcolor}
2   \definecolor{lightgray}{gray}{0.93} %defines my custom gray
3 \usepackage{listings}
4   \lstset{
5       language=Matlab,
6       numberstyle=\footnotesize,
7       firstnumber=1,
8       stepnumber=5,
9       numberfirstline=true,
10      breaklines=true,
11      backgroundcolor=\color{lightgray},
12      showspaces=false,
13      showstringspaces=false,
14      showtabs=false
15  }
16 \renewcommand{\lstlistingname}{Code Snippet}
17 \renewcommand{\lstlistlistingname}{List of \lstlistingname s}

```

Using this environment looks something like the following. Note that the label and caption syntax is different from the other environments we've looked at. We also define the file type here - it must be an m-code file. Live scripts (.mlx) will not properly import.

```

1 \lstinputlisting[caption={Creative Caption}, label={mylabel}]{my_matlab
   _file.m}

```

Bibliography

- [1] Herbert, S., *Robert Cheseman (Holbein)*. Chris Beetles Gallery, London, England, www.chrisbeetles.com/gallery/animals/cats/robert-cheseman-holbein.html. Accessed 8 Nov. 2018.
- [2] Lamport, L., *LATEX: a document preparation system: user's guide and reference manual*. Addison-wesley. 1994.
- [3] Overleaf c/o Digital Science., London, England, https://www.overleaf.com/learn/latex/Positioning_of_Figures. Accessed 16 Feb. 2019.
- [4] Stutts, D.S., *Mechanical Engineering Systems Laboratory Manual*. Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, 25 Jan. 2019. 142-146.
- [5] Vigen, T., *15 Insane Things That Correlate With Each Other*. Spurious Correlations, tylervigen.com/spurious-correlations. Accessed 8 Nov. 2018.
- [6] Gratzner, G.A., *Practical LaTeX*. 2014.