

A Short L^AT_EX Introduction

Motivation and getting Started

Will Hossack

(long-term and unreformed L^AT_EX-hacker)

What is L^AT_EX

L^AT_EX (being Layman's-T_EX) is a text-formatting *mark-up* language,

1. Edit a *source* file containing your text and layout instructions.
2. Process (or *compile*) it using the command `latex` to give a *device independent* `dvi` file.
3. Result can be viewed on screen or printed.
4. Option to convert to Postscript (for printer), or PDF for Web.

T_EX designed and implemented by Donald Knuth to format his book series *The Art of Computer Programming*. \approx 1980.

L^AT_EX “*simplified*” front-end by Leslie Lamport, originally for production of computer manuals \approx 1985.

Updated to L^AT_EX 2 ϵ in 1993, then to L^AT_EX 3, but all still compatible with original.

Best testament to a “really well designed” piece of software!, is works **and** has lasted.

Why use L^AT_EX

Superficially L^AT_EX looks old, difficult, out-dated, and a bit of a dinosaur, but. . .

1. It formats mathematics **faultlessly!**
2. It has mathematically (and esthetically correct fonts.)
3. It implements all the correct rules of typography.
4. It can produce truly beautiful documents.
5. It works on **all** computers, and is **totally reproducible** on all computers.
6. Is as close to **BUG-FREE** as you will ever find.
7. Its **FREE SOFTWARE** in the truest sense.

It's the standard for all mathematics, physics and many computer science literature including all journals and most textbooks.

Think **HTML** but actually done properly, and you are getting the idea!

The very basic L^AT_EX document

Start off by creating a simple file, say `document.tex` that contains this:

```
\documentclass[a4paper,12pt]{article}
\begin{document}
\begin{center}
  \Large Text formatting here I come!
\end{center}
Using \LaTeX\ for simple text is very easy,
you do't even have to worry about getting the lines the same
length!
```

Ever paragraphs are just extra blank lines, this is really is very ``easy``.

```
\end{document}
```

Now lets see what we have to do to process and print it...

The very basic L^AT_EX document II

Simplest Processing:

1. `latex document` process the file with `latex` which will produce a file `document.dvi`.
2. `xdvi document` display the `document.dvi` on a X-screen so see if it is OK.
3. `dvips document` takes in `document.dvi` file and produces `document.ps` a printable Postscript file.
4. `lp document.ps` prints the file to the default printer.

You can miss out step 2, but you wont see what you have until it is printed.

Replace `latex` with `pslatex` to use real Postscripts fonts that frequently look better.

Italic and Bold

Simplest scheme is to use `\it` and `\bf` key inside `{}`.

and `{\it you}` really `{\bf must}` get this correct!

will give you

and *you* really **must** get this correct!

You can also locally change font size with with `\tiny`, `\small`, `\large`, `\Large`, `\LARGE` key words.

This is “old” way of changing fonts, it works well, and easiest to use, the *new* and *correct* is much more verbose.

Other Symbols

There are full range of symbols and accents, all controlled by keywords, for example `\pounds 23.45` gives £23.45 and `Schr\"odinger` gives Schrödinger.

There is also full support for non-English languages, including oriental, Arabic, and hieroglyphics! even *fantasy* languages (but we have not installed all of them!!!).

Adding Mathematical Characters

Two types of maths, In-line and Display.

Inline Mathematics: Simple enclose in \$ signs, so that

and the particle has velocity $v = u + \alpha t \dots$

will give:

and the particle has velocity $v = u + \alpha t \dots$

Note: The `\alpha` gives α , and since we are in *maths mode* then all variables are *maths italic* and the spacing is correct for mathematics.

In *maths mode* you have all Greek letters and masses of other mathematical symbols available.

Display Maths

Display maths equations appear centre, typically with equation numbers, so

```
\begin{equation}
  \vec{E} = \frac{1}{4\pi\epsilon_0}
  \frac{q}{r^2}\hat{k}
\end{equation}
```

will give you,

$$\vec{E} = \frac{1}{4\pi\epsilon_0} \frac{q}{r^2} \hat{k} \quad (1)$$

Note: that *superscripts* and *subscripts* automatically scale as to.

Brackets and Integrals

Brackets and matched with the `\left` and `\right` key and again scale, also `\int` behaves as you would expect.

So even fairly complex equations, like,

$$E_z = \int_{\text{ring}} dE_z = \left[\frac{1}{4\pi\epsilon_0} \frac{z\lambda}{(R^2 + z^2)^{3/2}} \right] \int ds \quad (2)$$

can be set with

```
\begin{equation}
  E_z = \int_{\text{ring}} dE_z
  = \left[ \frac{1}{4\pi\epsilon_0}
  \frac{z\lambda}{
  \left( R^2 + z^2 \right)^{\frac{3}{2}}} \right] \int ds
\end{equation}
```

Matching up `{` and `}` is vital, and can be *character building!*

Matrices, Arrays and Multiline Equations

This is a bit tougher, but the most common is the matrix,

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (3)$$

which can be easily set using the `\matrix` construct, being

```
\begin{equation}
  {\bf M} = \left[\matrix{a & b\cr
                       c & d}\right]
\end{equation}
```

There are *many other* constructs, see book, on-line references and ask you local T_EX-pert.

Document Type and Sections

There are three standard document *types*

1. `article` short document with sections and subsections (the most useful).
2. `report` long document with chapters, sections and subsections.
3. `book` extension of `report` with different page layout.

Most useful is `article` class, declared at the top

```
\documentclass[a4paper,12pt]{article}
```

means an `article` document, but using `12pt` font and `a4paper`.

The “default” is 10pt font on US “Letter” paper, which is rather small font, on paper we don’t have!!

Sections and Subsection

Assume `article`, then you have

1. `\section{Title of Section}` start of section.
2. `\subsection{Title of a Subsection}` start of a subsection,
3. `\subsubsection{Title of a Subsubsection}` start of a subsubsection.

Size of fonts and numbering is all automatic. You can add/remove sections and the numbering will automatically change.

There is also “*” variants which do not have numbers, so `\subsection*{Un-numbered subsection}` will format a subsection but *without* a number heading.

Number and title can also be automatically included in a *Table of Contents*, which it really get right!

Floating Bodies

There are object that will not appear *immediately* but at the next convenient place. Typically **figures**, **tables** and **footnotes**.

Footnotes: There are easy, just do,

```
.. which you can easily\footnote{With pages of manipulation!} show...
```

will add a *superscript* in the text and a footnote (in reduced font) at the foot of the page.

Figures: are a bit harder, work through:

```
\begin{figure}[htb]
  <body of figure, often a Postscript file>
  \caption{This is a figure.}
  \label{fig:importantfigure}
\end{figure}
```

which will add the figure *as soon as possible*, with specified caption and label, which you refer to.

You can then refer to the figure by

```
.... as shown in figure~\ref{fig:importantfigure}....
```

Where Floating Bodies go...

Floating figures etc format using “typographic rules”, which are normally correct,

1. Make figure smaller, large figures are difficult to place.
2. Relocate definition, move declaration forward of back input file.
3. Accept your lot accept that L^AT_EX has done the “best possible”.

If you use labels, footnotes you may have to run `latex 2` or sometimes 3 times to sort out all cross-references.

Note: You can also use `\label{name}` for equations, section, subsection etc, so you can refer to them be `name`. When you modify document, all the cross-references are still correct!

Adding extra features...

Add extra packages to standard L^AT_EX by using the `\usepackage`, which **must** go before `\begin{document}`, so

```
\usepackage{fullpage,epsfig,hyperref}
```

will add packages

1. `fullpage` local package to fill a A4 page with sensible margins.
2. `epsfig` add including of Postscript files.
3. `hyperref` add including of `http` hyperref with work in PDF documents.

There are many thousands of possible extensions.

Adding Postscript Files

The simplest recipe is:

```
\begin{center}
  \epsfysize 60mm
  \epsffile{MyPostsScriptFigure.eps}
\end{center}
```

which will centre `MyPostsScriptFigure.eps` scaled to 60 mm high. (can alternatively use `\epsfxsize = 100mm` so scale to 100 mm wide.)

Note: the Postscript file **must** be `eps`, with a `BoundingBox`.

Utilities that work easily are:

1. `xfig` to draw diagrams.
2. `gnuplot` to plot graphs.
3. `maple` for graphs and function plots.
4. `xv` to convert gif/jpegs

Most PC utilities do **NOT** produce legal eps files.

Adding hyperrefs

Adding external hyperrefs is very easy,

```
\href{http://en.wikipedia.org/wiki/LaTeX}{see here}
```

will add the link to the Wikipedia entry for L^AT_EX.

When you make a PDF file, it will have a clickable link that will open your Web browser.

This is fairly new addition and, and it success will depend on the version of the other software on the system.

It does work fine in the CPLab.

Making PDFs

There are two routes, (neither perfect!).

1. `pdflatex` `document` takes a `.tex` file and directly produces a `pdf`. Works well with equations but **does not** work with include `eps` figures.
2. `pslatex` `document` to produce a `dvi` file, then `dvipdf` `document` to produce the `pdf`.

Works with `eps` figures **but** some bold capital maths symbols lost (■ should be `\bf\Lambda`).

This integration is getting better with each software release.

RedThere is also utilities to produce `html` and `xml` for web use, but its still a bit crude.

Using macros

L^AT_EX has powerful **macro** scheme, its simplest use is for symbol substitution. For example if you wanted your *vectors* to be **bold**, underlined & overarrow and you are doing a lot of electromagnetism, we can define two local macros, `\vE` and `\ce` by

```
\newcommand{\vE}{\underline{\vec{\bf E}}}
\newcommand{\ce}{\frac{1}{4\pi\epsilon_0}}
```

so that

$$\underline{\vec{E}} = \frac{1}{4\pi\epsilon_0} \frac{q}{r^2} \hat{k} \quad (4)$$

would be formatted by

```
\begin{equation}
\vE = \ce \frac{q}{r^2} \hat{k}
\end{equation}
```

Macro can also take parameters, also can include `if` statements and perform calculations... so a *programming language*.

Breaking up your input

It is also very useful to breakup your input into a series of source files chained together with `\input` commands,

```
\documentclass[a4paper,12pt]{article}
\usepackage{fullpage,epsfig}
\begin{document}
  \input title
  \input abstract
  \input theory
  \input results
  \input excuses
\end{document}
```

where the actual text is `title.tex`, `abstract.tex` etc.

Very useful feature when you have a large document written by several people.

Summary

L^AT_EX is good for:

1. Highly mathematical document (nothing else comes close).
2. Large, complex documents with many sections, and potentially many authors.
3. Technical books (or thesis) with complex structure.
4. Documents that must be totally cross-platform.
5. Documents with optional sections/formats.

But not so good for simple letters, forms etc.

Books

1. L^AT_EX : a document preparation system ; use's guide and reference manual, Edition 2, *Leslie Lamport*, Addison-Wesley, Z253.4.L38 LAM
the reference book, but a little short on examples.
2. The L^AT_EX companion, Edition, *Mittleback & Goossens*, Addison-Wesley, Z253.4.L38 LAT
masses of examples/recipes, but not always a good explanation of "how it actually works".

Local Resources

Some local links

- <http://www.ph.ed.ac.uk/~wjh/tex/>
my local L^AT_EX page, with template.
- <https://www.wiki.ed.ac.uk/display/SSLCWiki/Home>
Physics Student Wiki (look under tutorials, where there is instruction of how to install in your laptop)