Spring 2013 LaTeX Workshop

Getting Started

LaTeX , at its core, is a computer language which is used to specify the layout of a page. Once the code is written, it can then be compiled to produce a pdf, dvi, or ps document. There are a number of applications which exist specifically to help write and compile LaTeX documents, but code can be written on any text editor and compiled directly.

The first step to compiling your own code is installing a LaTeX distribution, which includes the actual programs used to compile your code into a .pdf document (or a .dvi, or a .ps document). With this is installed, you LaTeX in any text editor (such as notepad, vim, emacs, gedit, etc.), and build it into a pdf with the command 'pdflatex'. Even though all you need is the distribution and a basic editor, it is typically easier to learn on an editor specifically suited to LaTeX .

The installation process can vary depending on your operating system, but some of the more popular options are described below:

Windows

Miktex is an excellent LaTeX distrubution for the Microsoft Windows operating systems, which is easily installed and manages add-on packages for you as needed. It can be found here: `http://miktex.org/`.

For the editor, TeXnic Center (`http://www.texniccenter.org/`) is a LaTeX development suite which includes references, syntax highlighing, and preview options. When starting up TeXnic Center for the first time, it may ask you to tell it where your Miktex files are. This will typically be C:\Program files\MiKTeX\miktex\bin .

Mac OS X

MacTeX (`http://www.tug.org/mactex/2011/`) is the most popular LaTeX distribution on OS X, and it includes an excellent editor (TeXshop). MacTeX is a large program which is easy to set up and full of useful features.

Linux

The most popular distribution for Linux is TeXLive, which can be installed in a variety of ways depending on your specific Linux distribution (in Ubuntu: 'sudo apt-get install texlive').


TeXWorks is a popular and easy to set up LaTeX editor which is compatible with most popular Linux distributions. There are also a number of plugins for editors such as vim or emacs.

Note that there are numerous online resources for learning LaTeX (one of the most comprehensive and well written can be found here: `http://tobi.oetiker.ch/lshort/lshort.pdf`). If you run into trouble while installing, visit your particular distribution's website or try googling you're problem.

Compiling Your First Document

Once you have a LaTeX distribution installed, you're ready to compile your first document. Fire up any text editor, and write the following:

```
\documentclass{article}

\begin{document}

hello world! $e^{i \pi} + 1 = 0$

\end{document}
```

Then save this file as hello.tex. If you're using an editor built for LaTeX , there should be a compile button, which will automatically turn your code into a pdf. To compile manually, pull up a command line, navigate to your file, and run the command 'pdflatex hello.tex'. Either way, you should end up with a pdf containing only the line:

hello world! $e^{i\pi} + 1 = 0$

LaTeX Syntax

Looking at the above example, a few details stand out. The source code is a mix of text and formatting commands, and the math is written in between $ signs. A command in LaTeX typically is typically begun with a backslash, and arguments are passed in brackets (with options in square brackets) like so:
`\command[option]{argument1}{argument2}`. For example, `\frac{a}{b}` produces $\frac{a}{b}$.

The commands before the `\begin{document}` statement are known as the preamble and is where you place formatting options to change the appearance of your document. For example, the `\documentclass{article}` command tells LaTeX that we want to use the built in class 'article'. A class sets basic formatting options, such as fonts, margins, formatting for section headings, additional packages, spacing, etc. The 'article' class is very flexible and easily extensible. Other options include 'book', 'report', and 'beamer' (for presentations).


The part of the document between the `\begin{document}` and `\end{document}` lines is the body. This is where the actual content goes which will make up the document. Text is divided up into regular mode and math mode, and math can be either inline or in display mode. By default, text is assumed to be non-math. To format a mathematical expression, surround it by dollar signs or by `\( ... \)` to fit it in the current line, or surround it by double dollar signs or `\[ ... \]` to display it in its own line. For example,

`$ e^x = \sum_{k \geq 0} \frac{x^k}{k!}$` will compile to $e^x = \sum_{k\geq 0} \frac{x^k}{k!}$, while
`$$ e^x = \sum_{k \geq 0} \frac{x^k}{k!}$$` will come through as

$$e^x = \sum_{k\geq 0} \frac{x^k}{k!}.$$

Another important aspect of LaTeX is its handling of whitespace. The code:

```
\LaTeX         treats multiple      spaces
        as       just   a single  space,  and
  multiple blank        lines as    a   single     one    .
```

is typeset as

LaTeX treats multiple spaces as just a single space, and multiple blank lines as a single one.

A paragraph break is indicated either by a pair of backslashes or at least one blank line between blocks of text.

Adding packages

While the 'article' class is useful for most situations, and is easily extended to fit many situations, there are other classes specially suited for different tasks (notably: beamer, for presentations). Changing the formatting of an entire document (for example, for submission to a journal) is often as simple as changing the class.

Where the class defines some of the global specifications for the document, packages generally add functionality or environments to a document. For example, this document (whose source code can be found at `http://www.math.ufl.edu/gma/textalk`), uses the package 'fancyhdr' to make nice and customizable headers and footers, and 'hyperref' to embed hyperlinks.

For another example, the packages 'amsmath', 'amsthm', 'amssymb' provide additional mathematical symbols and theorem-like environments, which can be specified in the preamble. To see how this works, try compiling the following code. Anything written after a % sign is a comment and won't affect the code, which can be useful for explaining what your code does (to youself or someone else).


```
\documentclass{article}

\usepackage{amsmath,amssymb,amsthm}

\newtheorem{defn}{Definition}  % I called them defn and thm
\newtheorem{thm}{Theorem}      % for shorthand

\begin{document}

\begin{defn}
  Let $n,k \in \mathbb{Z}^+$ with $n \geq k$. Then define
  $$ \binom{n}{k} = \frac{n!}{k!(n-k)!}.$$
\end{defn}

\begin{thm}[Binomial Theorem]
  For all $x \in \mathbb{R}$ and $n \in \mathbb{Z}^+$,
  $$ (x+1)^n = \sum_{k = 0}^n \binom{n}{k} x^k.$$
\end{thm}

\begin{proof}
  Left as exercise.
\end{proof}

\end{document}
```

You should end up with something like:

Definition 1. *Let $n, k \in \mathbb{Z}^+$ with $n \geq k$. Then define*

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Theorem 1 (Binomial Theorem)**.** *For all $x \in \mathbb{R}$ and $n \in \mathbb{Z}^+$,*

$$(x+1)^n = \sum_{k=0}^{n} \binom{n}{k} x^k.$$

Proof. Left as exercise. □

More information about various packages and useful tips can be found at `http://en.wikibooks.org/wiki/LaTeX/`, and a more complete (and better written) introduction can be found at `http://tobi.oetiker.ch/lshort/lshort.pdf`.