Getting Started with LaTeX

The CS 299 Staff

January 14, 1998

1 Introduction

Why typsetting is important

We're requiring you to typeset your assignments for two reasons. The selfish reason is that it saves us time trying to decode your penmanship (or pencilmanship). But we're doing it for your sake, too: as a member of the world's scientific community, you're going to have to know how to communicate your ideas to others, and you might as well learn how to do that now, when you've got some patient TA's willing to help you out.

The intended audience for this handout is people who have never used LaTeX before.

What is LaTeX?

LaTeX is a customized version of the TeX program. The latter was written about twenty years ago by the illustrious computer scientist Donald Knuth. If you've never heard of him, you will, assuming you continue on in computer science. He's famous for *many* things, including a series of (now somewhat dated) books on algorithms and the Knuth-Morris-Pratt string-matching algorithm[1]. Unfortunately, though TeX is powerful, it's kind of a bear to use, which is where LaTeX comes in. LaTeX is, to paraphrase the author, an attempt to convert the highly-tuned racing car of TeX into a family sedan—more comfortable to use, though perhaps with fewer of the bells and whistles. Trust us that unless you've got a lot of time on your hands, you probably want to stick with LaTeX.

But why use LaTeX rather than, say, Framemaker or Word? Unfortunately, the answer's sort of like the answer to "why Windows?": it's a standard, for better or for worse. If you're collaborating on a scientific paper with an academic colleague in Spain or Canada or New Jersey, it's more likely that your colleague will know LaTeX than any other typesetting program. In other words, if English is the *lingua franca* of the world's scientific community, LaTeX is what the world's scientific community uses to typeset its ideas. Also, it's a pretty sure bet that since LaTeX is free, it'll be installed on any Unix environment you'll ever work in. It's installed, of course, on Andrew.

What you'll need know

We believe that the best way to learn LaTeX is by example, so along with this document you should be receiving a sample LaTeX document and the resulting output. The following section describes how exactly to create the output from the input.

2 Mechanics

Here's exactly what you'll have to do to generate a typeset document using LaTeX:

[1]the "Morris" is our department head, Jim Morris

1. Create your LaTeX document, naming it with the extension `.tex`. Let's say you call it `pancake.tex`. yitem Type `latex pancake.tex` at the Unix prompt. This creates a file called `pancake.dvi`, assuming LaTeX found no errors in your file (see below for instructions on how to deal with errors.)

2. Preview the document on the screen by typing `xdvi pancake.dvi`. Repeat steps 2-4 until satisfied.

3. Convert the document to Postscript by typing `dvips -o pancake.ps pancake.dvi`.

4. Print `pancake.ps` by typing `lpr -Pwean-1 pancake.ps` (substitute the appropriate printer name in place of `wean-1`).

3 LaTeX syntax

Words are separated by one or more spaces. Paragraphs are separated by one or more blank lines. The output is not affected by adding extra spaces or extra blank lines to the input file.

Special characters

\ All formatting commands are prefaced by a \ to distinguish them from text.

`%` Anything after a `%` to the end of the line is a comment, ignored by LaTeX.

{ and } The curly braces are used for scoping (to delimit parameters, for example).

`$` Equations to appear within a paragraph are begun with ended with dollar signs.

`&`, `#`, and _ These characters also have special meaning in LaTeX.

If you actually want one of these characters to appear, you can just type a backslash in front of them. You can get a percent sign with `\%`, for example. (The backslash (\) is an exception; we suggest you not worry about that yet.)

Environments and scoping

To italicize a portion of text in a WYSIWYG word processor, you highlight the chunk of text you want to italicize, and then tell the word processor to italicize the text. In LaTeX you indicate the scope of the italicize command with curly brackets.

`Here is some {\em emphasized} text.` `Here is some {\bf boldface} text.`	Here is some *emphasized* text. Here is some **boldface** text.

Other LaTeX commands, called *environments*, delimit their scope with `\begin{}` and `\end{}`. For instance, the `center` environment is used to center a paragraph.

`\begin{center}`
`Here is some centered text.`
`\end{center}`
	Here is some centered text.

Mathematical expressions

There are two ways of putting a mathematical formula in your document. For short expressions which fit inside text, use a dollar sign to begin and end the expression.

`The answer is $x=y^7$.`	The answer is $x = y^7$.

For more involved expressions, you'll want to use the `eqnarray` environment.

```
\begin{eqnarray}
\sum_{i=0}^n i = \frac{n (n+1)}{2}
\end{eqnarray}
```

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2} \qquad (1)$$

Here's where the power of LaTeX starts to show. LaTeX assigns a number to each equation you place within an `eqnarray` environment. If you want, you can assign a label (a character string) to an equation, so that later in the document you can refer back to the equation without having to know exactly what number LaTeX assigned to it. You accomplish this with the `label` and `ref` keywords.

```
\begin{eqnarray}
\sum_{i=0}^n i = \frac{n (n+1)}{2}
\label{little-gauss}
\end{eqnarray}

In equation \ref{little-gauss},...
```

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2} \qquad (2)$$

In equation 2,...

NB: If you do this, realize that LaTeX is going to need to make multiple passes over your document to reconcile `label`'s and subsequent `ref`'s. LaTeX will tell you about this with a message like:

```
LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.
```

Simply run LaTeX again on the same document and the second time the numbers hould be ok.

There's no sense being exhaustive here when there's a perfectly good reference available, namely *LaTeX: A Document Preparation System* by Leslie Lamport, which should be on reserve in the E&S library.

The magic of verbatim

LaTeX normally ignores the indentation and line breaks in your source file, preferring to reformat your paragraphs to fill the margins. If you don't want this behavior — for instance, if your document includes a program listing, where indentation and line breaks are important — then use the `verbatim` environment. All the characters you type between `\begin{verbatim}` and `\end{verbatim}` are rendered exactly the same way in the output, including spaces, newlines, and LaTeX special characters. In addition, the font changes to a monospaced ("typewriter") font, so that all characters have the same width.

```
\begin{verbatim}
int factorial (int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial (n-1);
}

\end{verbatim}
```

```
int factorial (int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial (n-1);
}
```

The `verbatim` environment always starts a new line. To avoid this, a short form of `verbatim` is also available, called `\verb`. Its argument is bracketed not by curly braces, but by any two identical characters. For example, `\verb|x\y|` becomes `x\y`. If you're having trouble persuading LaTeX to print a particular character, you may find `\verb` useful.

4 Dealing with errors

LaTeX is quite chatty. Here is the transcript of a *successful* (i.e. no errors) run on a four-page file:

```
% latex updates.tex
latex updates.tex
This is TeX, Version 3.1415 (C version 6.1)
(updates.tex
LaTeX2e <1994/06/01>
(/usr/local/lib/texmf/tex/latex2e/base/latex209.def
Entering LaTeX 2.09 compatibility mode.
(/usr/local/lib/texmf/tex/latex2e/base/tracefnt.sty)
(/usr/local/lib/texmf/tex/latex2e/base/latexsym.sty))
(/usr/local/lib/texmf/tex/latex2e/base/article.cls
Document Class: article 1994/06/02 v1.2s Standard LaTeX document class
(/usr/local/lib/texmf/tex/latex2e/base/size11.clo))
No file updates.aux.
(/usr/local/lib/texmf/tex/latex2e/base/Ulasy.fd)

[1]
Overfull \hbox (17.00584pt too wide) in alignment at lines 86--94
 □□□ □

Overfull \hbox (10.30635pt too wide) in paragraph at lines 94--98
\OT1/cmr/m/n/10.95 The in-qual-ity comes from an ap-pli-ca-tion of Jensen's in-
equal-ity: if $□□ \OML/cmm/m/it/10.95 q\OT1/cmr/m/n/10.95 (\OML/cmm/m/it/10.9
5 x\OT1/cmr/m/n/10.95 ) = 1$, then $□ □□ \OML/cmm/m/it/10.95 q\OT1/cmr/m/n/1
0.95 (\OML/cmm/m/it/10.95 x\OT1/cmr/m/n/10.95 )\OML/cmm/m/it/10.95 r\OT1/cmr/m/
n/10.95 (\OML/cmm/m/it/10.95 x\OT1/cmr/m/n/10.95 ) \OMS/cmsy/m/n/10.95 ^^U

LaTeX Warning: Reference `eq:B' on page 2 undefined on input line 124.

[2] [3] [4] (updates.aux)

(see the transcript file for additional information)
Output written on updates.dvi (4 pages, 15244 bytes).
Transcript written on updates.log.
%
```

You'll probably have to go through a few iterations of the edit-LaTeX cycle before you finally get something LaTeX accepts with no errors. When you get an error, you'll usually be told the line number, and you'll be shown the line number, with a newline inserted after the offending string:

```
! Undefined control sequence.
l.21 \dddddef
             \doubleexph{\sum_s \hmua \hmub e^{-s\hmua - (k-s) \hmub}}
?
```

At the question mark, your best bet is to press 'x' to stop LaTeX completely, although you could press 'r' to get LaTeX to plow ahead and ignore all further errors—though the end result will probably look nothing like what you had in mind.

5 More information

1. Lamport, Leslie. *LaTeX: A Document Preparation System.* Addison-Wesley, 1986.

   The official LaTeX reference. It's on reserve at the E&S library here. If you're serious about learning LaTeX, you might want to buy it.

2. MIT SIPB. *Inessential Latex*, 1995.
   `http://www.mit.edu/afs/sipb/project/doc/latex/guide.PS`

   An excellent introductory tutorial, with a reference showing the mathematical symbols that LaTeX can generate. Some of the instructions in this document are specific to MIT's computer system.

3. Latex Help 1.1
   `http://www.emerson.emory.edu/services/latex/latex_toc.html`

   A comprehensive reference manual to LaTeX commands and environments, indexed by command name. Useful if you want to know the parameters to a particular command.

For links to these and other references about learning and using LaTeX, check out the playground at our Web page, `http://www.discretemath.com/`