Intro to LaTeX, Part II: Math, BibTeX, and Customization

David Diez

OpenIntro
openintro.org

Outline

- Mathematics in LaTeX
- BibTeX: bibliographies in LaTeX
- Building your own commands and environments
- Miscellaneous tips

Guide to LaTeX

The book *Guide to LaTeX* offers a very nice introduction, and we will closely follow some of the examples in these chapters in this class:

7 math

11,12 BibTeX, and

10 custom commands and environments.

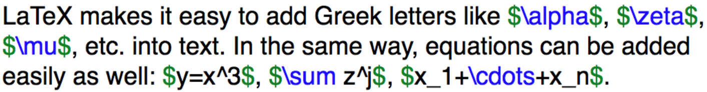
Math in LaTeX

We will cover several aspects of the mathematics environments offered in LaTeX.

- Basic mathematics in text
- Different equation environments
- Mathematical symbols
- Mathematical expressions
- Accenting and modifying text
- Automatic sizing of bracket symbols
- Text in mathematical equations
- Arrays and matrices

Inserting math into text

LaTeX makes it easy to add Greek letters like α, ζ, μ, etc. into text. In the same way, equations can be added easily as well: $y = x^3$, $\sum z^j$, $x_1 + \cdots + x_n$.



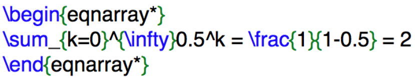
```
LaTeX makes it easy to add Greek letters like $\alpha$, $\zeta$,
$\mu$, etc. into text. In the same way, equations can be added
easily as well: $y=x^3$, $\sum z^j$, $x_1+\cdots+x_n$.
```

The \$ signs tell LaTeX when to switch into or out of math model. For instance, to create α above, type `$\alpha$`.

How can we create β?

Equation array

Some equations are long and should be on their on lines. In such a case, use the eqnarray or eqnarray* environment:



```
\begin{eqnarray*}
\sum_{k=0}^{\infty}0.5^k = \frac{1}{1-0.5} = 2
\end{eqnarray*}
```

The result in LaTeX for eqnarray*:

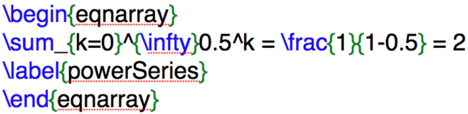
$$\sum_{k=0}^{\infty} 0.5^k = \frac{1}{1-0.5} = 2$$

Equation referencing

Just like tables and figures, equations can be referenced. Use `eqnarray` (no asterisk) to add an equation number:

$$\sum_{k=0}^{\infty} 0.5^k = \frac{1}{1-0.5} = 2 \qquad (1)$$

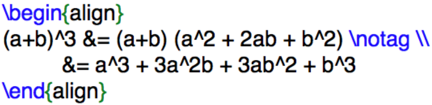
`\label{powerSeries}` can be put inside the equation array and then be referenced via `\ref{powerSeries}`.



```
\begin{eqnarray}
\sum_{k=0}^{\infty}0.5^k = \frac{1}{1-0.5} = 2
\label{powerSeries}
\end{eqnarray}
```

Aligned equations

Another environment, align (and align*) are handy for aligning multiline equations.



```
\begin{align}
(a+b)^3 &= (a+b) (a^2 + 2ab + b^2) \notag \\
        &= a^3 + 3a^2b + 3ab^2 + b^3
\end{align}
```

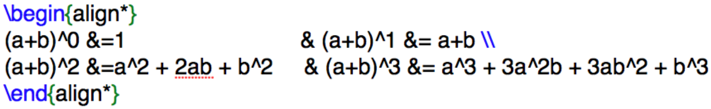
Result:

$$\begin{align}(a+b)^3 &= (a+b)(a^2+2ab+b^2) \notag \\ &= a^3+3a^2b+3ab^2+b^3 \tag{2}\end{align}$$

The \\ command creates a line break. The command \notag was used to suppress the equation number of the first line, which requires the amsmath package. (Q: We have an equation number. What should I have included in the code?)

Multiple alignments

The `align` environment permits several alignments:



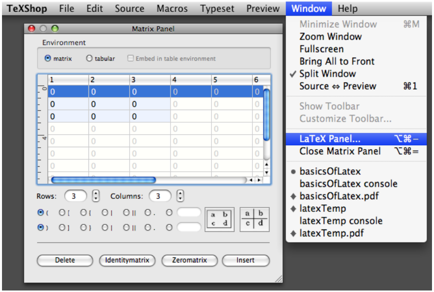
```
\begin{align*}
(a+b)^0 &=1                       & (a+b)^1 &= a+b \\
(a+b)^2 &=a^2 + 2ab + b^2     & (a+b)^3 &= a^3 + 3a^2b + 3ab^2 + b^3
\end{align*}
```

outputs

$$\begin{aligned} (a+b)^0 &= 1 & (a+b)^1 &= a+b \\ (a+b)^2 &= a^2+2ab+b^2 & (a+b)^3 &= a^3+3a^2b+3ab^2+b^3 \end{aligned}$$

Mathematics and symbols

It is a little difficult to learn all the math syntax and a good help source is the LaTeX and Matrix Panels:



The Matrix Panel is especially useful since matrices can require a lot of writing. The LaTeX panel is handy as a quick reference.

Some symbols

Here is a very small subset of the symbols available in LaTeX.

$\leftarrow$	`$\leftarrow$`	$\Leftarrow$	`$\Leftarrow$`	$\leftrightarrow$	`$\leftrightarrow`
$\geq$	`$\geq$`	$\neq$	`$\neq$`	$\not\in$	`$\not\in$`
∂	`$\partial$`	$\oint$	`$\oint$`	∇	`$\nabla$`
$\bigcap$	`$\bigcap$`	$\bigcup$	`$\bigcup$`	$\cap$	`$\cap$`
$\subset$	`$\subset$`	$\supseteq$	`$\supseteq$`	$\not\supseteq$	`$\not\supseteq`
$\bigodot$	`$\bigodot$`	$\bigotimes$	`$\bigotimes$`	$\oplus$	`$\oplus$`
$\clubsuit$	`$\clubsuit$`	$\perp$	`$\perp$`	$\vdash$	`$\vdash$`

For a searchable PDF with thousands of symbols, see

www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf

Also see the LaTeX Panel (under the menu item Window).

Character modifications

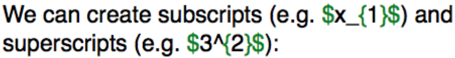
Text and symbols in math mode can also be modified.

Regular		Modified		Accents	
`$R$`	R	`$\mathbb{R}$`	$\mathbb{R}$	`$\tilde{R}$`	$\tilde{R}$
`$A$`	A	`$\mathcal{A}$`	$\mathcal{A}$	`$\widetilde{A}$`	$\widetilde{A}$
`$x$`	x	`$\mathbf{x}$`	$\mathbf{x}$	`$\bar{x}$`	$\bar{x}$
`$p$`	p	`$\mathit{p}$`	p	`$\hat{p}$`	$\hat{p}$
`$X$`	X	`$\mathrm{X}$`	X	`$\widehat{X}$`	$\widehat{X}$

Two other accents: $\dot{x}$ and $\ddot{x}$ via `$\dot{x}$` and `$\ddot{x}$`.

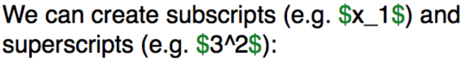
Subscripts and exponents

We can create subscripts (e.g. x_1) and superscripts (e.g. 3^2):



```
We can create subscripts (e.g. $x_{1}$) and
superscripts (e.g. $3^{2}$):
```

When the subscript is a single character, then it is acceptable to omit the curly braces. That is, the following is equally acceptable for the text above:

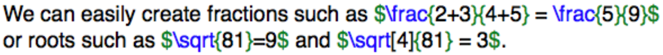


```
We can create subscripts (e.g. $x_1$) and
superscripts (e.g. $3^2$):
```

If more than one character is in the sub/superscript, braces are necessary to avoid problems: `$2_10$` outputs 2_10. Sub and superscripts can be used simultaneously: x_{ij}^2.

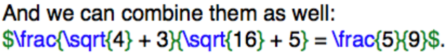
Fractions and roots

We can easily create fractions such as $\frac{2+3}{4+5} = \frac{5}{9}$ or roots such as $\sqrt{81} = 9$ and $\sqrt[4]{81} = 3$.



```
We can easily create fractions such as $\frac{2+3}{4+5} = \frac{5}{9}$
or roots such as $\sqrt{81}=9$ and $\sqrt[4]{81} = 3$.
```

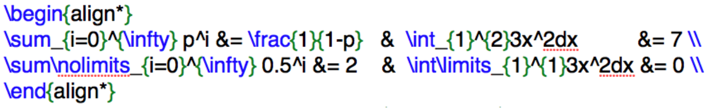
And we can combine them as well: $\frac{\sqrt{4}+3}{\sqrt{16}+5} = \frac{5}{9}$.



```
And we can combine them as well:
$\frac{\sqrt{4} + 3}{\sqrt{16} + 5} = \frac{5}{9}$.
```

Sums and integrals

We can also create sums and integrals:



```
\begin{align*}
\sum_{i=0}^{\infty} p^i &= \frac{1}{1-p}   &  \int_{1}^{2}3x^2dx        &= 7 \\
\sum\nolimits_{i=0}^{\infty} 0.5^i &= 2    &  \int\limits_{1}^{1}3x^2dx &= 0 \\
\end{align*}
```

which results in

$$\sum_{i=0}^{\infty} p^i = \frac{1}{1-p} \qquad \int_{1}^{2}3x^2dx = 7$$

$$\sum\nolimits_{i=0}^{\infty} 0.5^i = 2 \qquad \int\limits_{1}^{1}3x^2dx = 0$$

The commands \nolimits and \limits can be used to override the default displays of limits in LaTeX.

Practice

Produce the following result using the eqnarray* environment:

$$\sum_{i=0}^{n} p^{i} = \frac{1 - p^{n-1}}{1 - p}$$

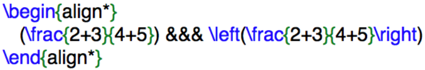
Some examples may be utilized in latexTemp.tex.

Sizing of Brackets

A small problem with bracket sizes is shown in the left equation, and this problem is fixed on the right.

$$(\frac{2+3}{4+5}) \qquad\qquad \left(\frac{2+3}{4+5}\right)$$

The coding for the expressions above



```
\begin{align*}
  (\frac{2+3}{4+5}) &&& \left(\frac{2+3}{4+5}\right)
\end{align*}
```

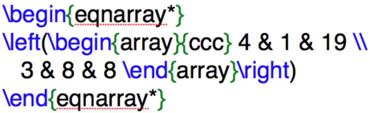
Generally we can use \left(, \left[, \left|, and \left\{ and their corresponding right brackets to create automatically sized brackets. These commands *must* be inside one of the equations environments and the left and right brackets must always be balanced.

Matrices

Matrices also can be made in LaTeX:

$$\left(\begin{array}{ccc} 4 & 1 & 19 \\ 3 & 8 & 8 \end{array}\right)$$

The code:



```
\begin{eqnarray*}
\left(\begin{array}{ccc} 4 & 1 & 19 \\
  3 & 8 & 8 \end{array}\right)
\end{eqnarray*}
```

The syntax for an `array` is the same as for `tabular` (a table).

Space and stacking

Space can be added in equations using \quad, and expressions can be stacked via \stackrel:

```
\begin{eqnarray*}
  E(X+Y) \stackrel{indep.}{=} E(X) + E(Y)
    \quad\quad
    Var(X+Y) \stackrel{indep.}{=} Var(X) + Var(Y)
\end{eqnarray*}
```

produces

$$E(X+Y) \stackrel{indep.}{=} E(X)+E(Y) \qquad Var(X+Y) \stackrel{indep.}{=} Var(X)+Var(Y$$

Why use BibTeX?

There are a number of very good reasons to use BibTeX instead of manual creation of bibliographies.

- Automatic creation of bibliographies.
- Easily change bibliographic styles.
- Identification of missing sources referenced in text.

How BibTeX works

There are three steps that LaTeX and BibTeX take to make a bibliography.

- When you typeset a document with citations (e.g. \cite{zotova}), LaTeX makes note of each citation.
- BibTeX takes this list and looks for each reference in a database of publications.
- Then we tell LaTeX to make the bibliography of all of those publications it found that we referenced.

The most time consuming part is initially building the database. After that, you can reference this same database over and over again and BibTeX becomes a breeze.

BibTeX material

- Creating your database
- Citing a reference
- Typesetting with BibTeX
- Building style files

Sample reference entry

We want to create a reference, similar to the following, for each of the item we want to cite.

@article{$\overbrace{\text{zotova}}^{label}$,
Author = {Elena Zotova and Charles D Woody and Ehud Gruen},
Journal = {Brain Research},
Pages = {66-78},
Title = {Multiple representations ... [etc etc].},
Volume = {868},
Year = {2000}}

Make up of a reference

Each reference needs a publication type (e.g. article, book), and each reference includes many fields. For instance, the following are **required** and *optional* fields of an article entry.

label: The reference label.

Author	**Journal**	**Title**
Year	*Volume*	*Number*
Pages	*Month*	*Note*

A formal list of the available publication types and also which items are required and optional for each type, see

http://www.image.ufl.edu/help/latex/entry_bibtex.shtml

An alternative

If you don't want to build your data base up in such a bare-bones manner, you might try

- BibDesk: Macs.
- JabRef: Macs, PCs, Linux.

Both of these programs are free and available online. Others exist, and I have only personally used BibDesk.

If you do manage your own...

Some things you should know if you do not use a program manager:

- Always include a label, which is how LaTeX identifies the entry.
- The entry (publication) type and field names are NOT cap-sensitive.
- Enclose the text for each field (e.g. the author names) in curly braces.
- You can add extra fields that are not listed and these will be ignored (e.g. if you add an Abstract field to a reference, BibTeX will just ignore it).

Special cases

Giving author names in a non-ambiguous form is sometimes difficult.

- Always type names as {Given Names Surnames} or {Surname, Given Names}.
- Anything enclosed in braces will be treated as a single item (e.g. Author = {Maria {San Martino}}).
- If there is more than one author, separate each author name by the word *and*. If *and* is part of someone's name, enclose their entire name in braces.
- You may add accents (e.g. Gödel via `G{\"o}del`).

Many other nuances exist. If you encounter a peculiar name, do a little online searching to see how best to put it into the data base.

Abbreviating journal names

Sometimes you don't want your entry to include the entire journal name. To shorten it, use the *string* entry type:

@string{JSS = {Journal of Statistical Software}}

These string entries must be defined in the database above where they are used.

Citing a reference

There are four commands that can be used.

- \cite{labelName} [referenceNumber], e.g. [1].
- \citet{labelName} Surname (year), e.g. Zotova et al. (2000).
- \citep{labelName} (Surname, year), e.g. (Zotova et al., 2000).
- \nocite{labelName} Not cited but will show up in bibliography.

The first and last work with the `uclathes` class. The second two are used in the `natbib` package (highly recommended for non-thesis papers).

Other commands in your document

The following two lines of code must be inserted at the place where the bibliography is to be added:

```
\bibliographystyle{yourStyle}
\bibliography{databaseName}
```

The style command can be moved higher (it doesn't matter). If you use the `natbib` package, then you must add it with the other packages:

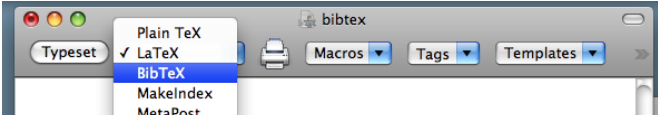
```
\usepackage{natbib}
```

I have not gotten this package to work with the UCLA thesis template.

Making the bibliography

If you have made your reference database, made citations, and inserted the bibliography commands in your text, then you are ready to create the bibliography. In TeXShop, there are a few simple steps to finish:

- Typeset your LaTeX document as usual.
- Change the Typesetting option from LaTeX to BibTeX:



- Typeset again with BibTeX.
- Return the Typesetting option back to LaTeX and compile **twice** more.

Building a style file

One of the big benefits of BibTeX is the ability to quickly change the bibliography style and within-text citations. To do this, we use the program custom-bib. Download it at

> http://www.ctan.org/tex-archive/help/Catalogue/entries/custom-bib.html

custom-bib has been included in the latexTemp zip file from the first class.

Building a style file

Open `latexTemp` > `custom-bib`, and open the `makebst.tex` file. To run the program,

(1) Open the file and typeset it.

(2) Type `YES` to the first question to get extra directions.

(3) Choose an appropriate file name (no need to add the extension).

(4) Answer each of the style questions.

(5) For the last question, *Finished!! ... Shall I now run this batch job? (NO)*, type `YES`.

Find and copy the file you named in step (3) with extension `.bst`. Put it in the folder with whatever files for which you will make a bibliography with this style or in your bibliography folder (however you reference it in your LaTeX document).

Practice

Open the `latexTemp.tex` file and go to the last section. Add a bibliography reference of \citet{victor}. Also add a reference with \citep{victor} and typeset (all four steps). What is the difference between your references? How would you use each in a paper?

Custom command material

- Counters
- Creating commands
- Creating environments

Existing counters

LaTeX uses counters (variables) to keep proper numbering.

- The following are counters used by LaTeX: `part`, `chapter`, `section`, `subsection`, `subsubsection`, `page`, `footnote`, `equation`, `figure`, and `table`. These counters correspond to their corresponding commands.
- Other counters are used for each level of enumerate: `enumi`, `enumii`, `enumiii`, `enumiv`.
- A few other LaTeX counters: `paragraph`, `subparagraph`, `mpfootnote`.

Create new counters

We may want to create our own counters for our own purposes. Maybe we have examples that we want numbered.

```
\newcounter{counterName}[inCounter]
```

This creates a new counter called `counterName`. The `inCounter` argument + brackets are optional and an `inCounter` is used to reset `counterName` whenever `inCounter` increments (e.g. `subsection` has "inCounter" `section`).

Modifying

We can modify existing or new counters.

```
\setcounter{counter}{n}
\addtocounter{counter}{n}
\stepcounter{counter}
\refstepcounter{counter}
```

The \refstepcounter command lets us reference our counter value if we follow it with a \label.

Printing

So we can create, modify, and reference counters. However, we also need to print counters in the document. We do so by calling the counter with one of the following commands:

\arabic{chapter} (4, Arabic number)

\Roman (IV, uppercase Roman numeral)

\roman (iv, lowercase Roman numeral)

\Alph (D, capital letter)

\alph (d, lowercase letter)

\fnsymbol (§, footnote symbol)

We will put counters to use in our custom commands and environments.

Simple command

Common statements, like $x_1, ..., x_n$ can be abbreviated using a new command.

```
\newcommand{\xvec}{x_1,\dots,x_n}
```

Inserting this command and then typing (later in the document) `$\xvec$`, we get $x_1, \dots, x_n$. If we forgot the dollar signs, we would be in trouble. We can resolve this by using an extra command:

```
\newcommand{\xvec}{\ensuremath{x_1,\dots,x_n} }
```

In the second definition, we left an extra space at the end, which helps prevent spacing problems. More elegant solution is to use the xspace package (see *Guide to LaTeX*, page 186).

Command with arguments

If we want to generalize our command, we add two arguments

```
\newcommand{\subvec}[2]{\ensuremath{#1_1,\dots,#1_{#2}}
}
```

We can create $y_1, \dots, y_m$ from `\subvec{y}{m}`.

Additional arguments can be created and are referenced via #n for the n^{th} argument. Optional default arguments can also be utilized (see *Guide to LaTeX*, page 188).

Generalization

The general framework of new commands is

\newcommand{\commandName}[n]{the commands}

where

- `commandName` is the name of the command,
- `n` is the number of arguments, and
- the arguments are referred to as #1, #2, . . . , #n in the commands.

To redefine a command that already exists, use \renewcommand with the same format as above.

Sample environment

Environments use begin and end tags (e.g. `itemize`). We only need define what happens at the \begin and \end tags. For example,

```
\newenvironment{example}
 {\small\textbf{Example.} \hspace{2mm}} % begin stuff
 {\\} % end stuff
```

Sample environment call:

```
\begin{example}
Modular addition works in mysterious ways: $2+2=1$ (mod 3).
\end{example}
```

Result:

Example. Modular addition works in mysterious ways: $2 + 2 = 1$ (mod 3).

General environment

Generally environments take the form

```
\newenvironment{environmentName}{begin stuff}{end stuff}
```

We can also declare that there will be `n` arguments.

```
\newenvironment{environmentName}[n]{begin stuff}{end stuff}
```

As before, we refer to the arguments as #1, . . . , #n in the begin stuff and end stuff.

To redefine an environment that already exists, use \renewenvironment with the same format as above.

Environment + counter

```
\newcounter{example}
\setcounter{example}{0}
\newenvironment{example}
  {\refstepcounter{example}\small
    \textbf{Example \arabic{example}.}\hspace{2mm}}
  {\\}
```

Organizer and time saver

The \include command is useful for long documents:

```
\include{otherDocName}
```

For instance, this presentation actually calls three separate documents: one for each big section. Thus I would not take time Typesetting parts of the document I was not working on while keeping organized:

```
\include{math/math} % "math" document in the "math"
folder
%\include{bibtex/bibtex}
%\include{comenv/comenv}
```

Wrap-up

After this class, you should have a general idea of

- using the math modes in LaTeX,
- creating bibliographies using BibTeX, and
- creating your own commands and environments.

Any questions?