Math Mode, Equations, and Environments

Lecture 2

David Kraemer & Caleb Leedy

October 16, 2014

Last time

- We installed engines to compile LaTeX onto our computers.
- We also installed TeX*Maker*, a helpful development environment.
- We started playing with simple documents, involving
  - document class,
  - package handling,
  - simple equations

Tonight's objectives

- We will get acquainted with the various forms of equation editing:
  - Math mode,
  - Display-line equations,
  - Equation environments,
  - Special mathematical objects
- We will do an exercise in dictation with the various components we have learned.
- Time permitting, we will also work on useful theorem formatting technique.
- In addition, we will talk about defining simple commands.

Preamble

- Let's set up a new document. We can start with the following code:

Preamble code

```
\documentclass{article}

\usepackage{amsmath, amssymb}

\begin{document}



\end{document}
```

- `amsmath` and `amssymb` are helpful math packages provided by the American Mathematical Society

Math mode

- In LaTeX, there are two main types of math typesetting options:
  - Math mode
  - Display mode
- Math mode is useful for in-line equations and symbols
- Display mode is useful for centered, numbered, eye-catching equations.
- Math mode typesetting is surrounded by a pair of $ signs:
- So, `$x+y+z=A$` will typeset into the equation $x + y + z = A$.
- When in math mode, you are able to use a plethora of math-related symbols, such as $\in$, $\unlhd$, and $\int$, that cannot be used outside of it.

Math mode

- To typeset a particular structure, you need to know the command for it.
- TEX*Maker* has an index of common math commands that you may insert.
- Otherwise, you can make use of the following helpful websites:
  - `http://faculty.cbu.ca/srodney/ShortSymbInd.pdf`
  - `http://detexify.kirelabs.org/classify.html`
- Detexify is a wonderful tool for looking up symbols by drawing them.
- Otherwise, you can use the reference sheet to look up symbols manually.

Useful math commands

Here is a very short list of useful math commands:

- `\beta` → β (In general, Greek works this way)
- `\frac{1}{2}` → $\frac{1}{2}$
- `\mathbf{A}` → $\mathbf{A}$
- `\{ \}` → $\{\}$
- `z \in \mathbb{R}` → $z \in \mathbb{R}$.
- `x^{2}` → x^{2}.
- `T_{5}` → T_{5}.
- `\int_{a}^{b}` → $\int_{a}^{b}$.
- `\sum_{i=1}^{n}` → $\sum_{i=1}^{n}$.

Try it out!

- Try to type the equation $\frac{d}{dx} f(x) = \gamma x^2$.
- Another example: $\lim_{x \to \infty} \frac{1}{x} = 0$.
- Another: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

Display mode

- Display mode works almost exactly the same as math mode.
- There are two ways to enter display mode:
  - `$$ equation $$`
  - `\[ equation \]`
- By convention, it is preferable to use `\[ equation \]`.
- Try displaying:

$$f(b) - f(a) = \int_a^b \frac{d}{dx} f(x) dt$$

Dictation example

Let's try typesetting the following paragraph from *Advanced Calculus*:

Sample

Let Σ be a finite piece of a surface. A *subdivision* Δ of Σ is a set of closed curves $\{C_k\}_1^n$ lying in Σ and dividing it into a set of n subregions of areas $\Delta\Sigma_k$, $k = 1, 2, \ldots, n$. The *diameter* of a region on Σ is the length of the largest straight line segment whose ends lie in the region ... Then the *surface integral of $P(x, y, z)$ over* Σ is

$$\iint_{\Sigma} P(x, y, z) d\Sigma = \lim_{\|\Delta\| \to 0} \sum_{k=1}^{n} P(\xi_k, \eta_k, \zeta_k) \Delta\Sigma_k$$

when this limit exists.

Matrices and other fun

- A different type of math typesetting involves matrix structures.
- Here, we need to use an environment such as
- `\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2}\end{pmatrix}` becomes

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2}\end{pmatrix}$$

- The & symbols indicate spatial separation alignment.
- The `\\` indicates a line break.
- There are several other types of matrix environments: `bmatrix`, `Bmatrix`, `vmatrix`, and `Vmatrix`
- Try them out!

Environments

- `pmatrix` is a special case of what is called in LaTeX-speak as environments.
- Environments are blocks of code of the form:

Environments

```
\begin{environment}

\end{environment}
```

- Essentially, these tell the LaTeX compiler that the nested code will have properties specified by the code for the `environment`.
- In the case of `pmatrix`, the code simply takes the environment defined by the `matrix` environment and slaps on parentheses.

Environments

- Every LaTeX document involves one basic environment: the `document` environment.
- The reason for this is because it applies basic standards to the underlying TeX engine.
- Environments are typically useful for adding formatting criteria.

The eqnarray environment

- Let's consider the eqnarray environment, which is handy for aligning multiple equations.
- It also auto-numbers equations in a list.

Example eqnarray

```
\begin{eqnarray}
a^2+b^2=c^2 \\
1 + 2 = 3
\end{eqnarray}
```

- This produces neatly-formatted equations with numbers on the right.
- To remove the equation numbers, replace eqnarray with eqnarray*

The `align` environment

- The `eqnarray` environment is nice, but it sometimes gets inconsistent.
- An alternative is the `align` environment, which allows you to force the alignment as you want using &.

Example `align`

```
\begin{align}
(a+b+c)^2 &= (a + (b+c))^2 \\
&= a^2 + 2a(b+c) + (b+c)^2 \\
&= a^2 + b^2 + c^2 + 2ab + 2ac + 2bc
\end{align}
```

- Again, replacing `align` with `align*` will suppress the line numbering.

Commands

- As opposed to environments, commands in LaTeX are typically less-broad and more focused in what they achieve.
- Commands are of the form `\command{arg1}{arg2}...{argN}`.
- Some useful examples are
  - `\textit{text}`, which yields *text*
  - `\textbf{text}`, which yields **text**
  - `\textsc{text}`, which yields TEXT
  - `\textsf{text}`, which yields text
  - `\textsl{text}`, which yields *text*
  - `\texttt{text}`, which yields `text`

Creating a command

- An easy first command is to replace `\mathbb{R}` command with a shorter one, since $\mathbb{R}$ pops up often.
- To add a command, go to the preamble of the document and add the following line:

```
\newcommand{\R}{\mathbb{R}}
```

- Now, simply typing `\R` will yield the same result.
- In general, command declarations will be of the form:

```
\newcommand{\command}[number of args]{command action}
```

- You can also redefine existing commands with the `\renewcommand{\command}{command action}`.