Math 291: Lecture 7

Dr. Fagerstrom

Minnesota State University Moorhead
web.mnstate.edu/fagerstrom
fagerstrom@mnstate.edu

March 28, 2019

Table of Contents

Outline

General Comments and Comments from Lab 6

- I have rearranged the remaining topics for the semester due to the winter cancellations. Basically, I thought topics like BibTeX needed to be covered sooner than the pushed-back schedule allowed for. So although we will still cover the remaining topics, they won't be in the order indicated on the syllabus.
-

Outline

Partitioning your document

The \section command from Beamer can be used in any LaTeX document, and there are additional sectioning commands based on the type of document.
The most common of these commands are as follows:

- `\part{name of part}`
- `\chapter[shorter name]{full name of chapter}`
- `\section{name of section}`
- `\subsection{name of subsection}`
- `\subsubsection{name of subsubsection}`
- `\paragraph{paragraph heading}`
- `\subparagraph{subparagraph heading}`

Partitioning your document

Notes:

- Parts, chapters, sections, subsections, and subsubsections are numbered within the document.
- Paragraphs and subparagraphs are not numbered, but, like the other categories, they are given a bold heading.
- 'Smaller' categories have smaller fonts used in the heading for that category.
- All these commands are built into the article document class, with the exception of the chapter command, which is used in the report document class.

Outline

Title Page

- Beamer files usually have title pages, but so do other documents.
  - Put the following in the preamble:

    ```
    \title{title text}
    \author{name (or add \\ name 2, etc)}
    \date{enter desired date}
    ```

  - Note that if you don't supply a date, LaTeX automatically uses today's date.
  - All of the above merely gives LaTeX information to put into a title page, but doesn't create the title page itself. To do that in most document types, type \maketitle just after the begin{document} command (or wherever you want the title page to appear).

Table of Contents

- Once you have sections or other divisions that could be put into a table of contents, it makes sense to have a table of contents.
- To create a table of contents, type the following command at the beginning of your document:
  \tableofcontents
- Note: As was the case with Beamer, you have to build twice to get this to populate correctly (once to populate the .log file, once to read the now-populated .log file).
- See the ExampleDocWks7and8 document to see an example of how the sectioning and table of contents commands work.
- Note that if you want the table of contents on its own page, you may need to use a \newpage command.

Outline

Building Bibliographies

- LaTeX has a related program called BibTeX that will automatically build bibliographies, including their references within a document.
- BibTeX requires both a *separate* file with a .bib extension and using the BibTeX build command as well as LaTeX.

Creating .bib entries

There are several different types of bibliographic entries. Standard ones are:

- article (requires: author, title, journal, year)
- book (requires: author or editor, title, publisher, year)
- inbook (requires: author or editor, title, chapter and/or pages, publisher, year)
- misc (optional: author, title, howpublished, month, year, note)

For a fuller list, and other detailed information, see:
http://bibliographic.openoffice.org/bibtex-defs.html

Creating the .bib file

The BibTeX entries look somewhat complicated, but mathematicians, being who we are, have worked to make life easier...
One nice way to get .bib entries in the correct format is to obtain them from MathSciNet:
MathSciNet is a fairly comprehensive reference archive for articles in mathematical journals. (Use it on campus – it is not free!)
To start, first download and open the file 2019-291-Week7ExampleBib.bib from my webpage.

Creating the .bib file

(1) Go to MathSciNet: http://www.ams.org/mathscinet/search.html

(2) Do a **publications** search by typing in the name of your favorite author (Aryal, Ashok).

(3) Check a couple of boxes.

(4) In the pull-down menu, change it to: Citations (BibTeX).

(5) Choose: Retrieve Marked.

(6) Open 2019-291-Week7ExampleBib.bib

(7) Copy and paste these into the example .bib file. (Presumably at the end of it.)

Hints for the .bib file

The Citation Key

- The Key comes after the typed entry. @ARTICLE{CitationKey...
- MathSciNet gives each article a unique key, but it is not very handy reference.
- Typically, we will change this to a tag that is easier to remember and reference, as I did in the example .bib file. For example, Aryal2019 and AryalThesis.

Any field in a .bib reference that is neither required nor optional is ignored - so you may include whatever you want in your .bib entries.

Using the .bib file

To generate a bibliography, at the very end of your .tex file (just prior to the `\end{document}`), add the following commands:

`\bibliographystyle{plain}`

`\bibliography{NameOfBibFile}` (Do not include the .bib extension in the file name.)

Running BibTeX

- Depending on your LaTeX interface, you may need to build four times:
  1. Run LaTeX (regular build)
  2. Run BibTeX (from the pull-down menus: Build - Current File - BibTeX)
  3. Run LaTeX
  4. Run LaTeX

Running BibTeX, continued

- Note that your bibliography at this point will have no content.
- In the bibliography in your final document, only those sources actually cited appear.
- Your .bib file can contain (and often does) any paper you've ever referenced. You just continue adding references as needed. (Note: This is one realization of the 'mathematicians are lazy' rule.)

Using the .bib file

- To refer to one of our bibliographic references in a document, we use the command:
- `\cite{CitationKey}`
- Practice by adding a citation referencing the sources Goyt2016 and AryalThesis from the .bib file.

Bibliography styles

- Note that our bibliography, using this style, is alphabetized.
- Common bibliography styles are:
  - plain
  - alpha
  - unsrt
  - abbrv
- There are other styles. Google it!
- You might also want to investigate the commands `\begin{thebibliography}` and `\bibitem` for use without a .bib file.
- But note that most mathematicians just use their lifetime-global list (again, the 'mathematicians are lazy' rule...).

Bibliographies

- As mentioned earlier, only references that are actually cited appear in the Bibliography.
- You can get an individual reference to appear without citing it by using (typically at the end of your file)

  ```
  \nocite{key}
  ```

- If you want to see every item in your Bibliography (sometimes used to find a particular item in your lifetime-global list when it gets very long), you can use

  ```
  \nocite{*}
  ```

Bibliographies

- What do you do if you want to reference a history paper (not in MathSciNet)?
- One method is to use the information in the web resource linked to earlier in this presentation.
- Another method is to find a similar type of document in MathSciNet (for example, to cite a journal article in history, find a journal article in math), then
- Copy the citation to you .bib file and change the appropriate bits to the correct information for the history paper.
  - change the author name
  - change the journal name
  - change the date
  - etc.
- BibTeX will take that new data and appropriately format it.

Outline

User-Defined Commands

In addition to all of the built-in LaTeX commands, you can also create your own commands (or shortcuts).

- The syntax for defining a command is:
  \newcommand{\name}[#args][opt]{def}
  Note: LaTeX will not allow you to redefine a command that has already been defined internally.
- **Basic Example:** \newcommand{\di}{\displaystyle}
- When placed in the preamble, the command is globally defined (applies to the entire document).
- When placed within an environment, it is defined only within that environment.
- When placed elsewhere in the body of a document, it can only be used from then on.

Required Arguments of New Commands

- The "#args" part of the newcommand syntax indicates the number of arguments that are required to be supplied when using the command (when the new command is used, each argument should be put within a separate "{}").
- Each argument will be referred to separately in the definition of the command by using: #1, #2, etc.
- The command
  \ensuremath
  ensures the command will **always** be carried out in math mode (whether you call the command inside a math mode or not).

Examples of Commands with Arguments

- Code:

```
\newcommand{\repdec}[1]
{\ensuremath{0.\overline{#1}=\frac{#1}{99}}}
```

- Then call your new command by entering:
  `$\repdec{43}$`
- Output is: $0.\overline{43} = \frac{43}{99}$
- Notice that this is a command with a single argument (input).

Examples of Commands with Arguments

- Code for a command requiring 4 inputs:

```
\newcommand{\compfrac}[4]
{\ensuremath{\frac{\frac{#1}{#2}}{\frac{#3}{#4}}}}
```

- Then enter `\compfrac{1}{2}{3}{4}`
- Output: $\frac{\frac{1}{2}}{\frac{3}{4}}$

Example of a command with an optional argument

- The \newcommand also allows you to define commands with one *optional* argument (an argument that is available for use but not absolutely required).
- Example code:

```
\newcommand{\subvec}[3][x]{#1_{#2}, \ldots, #1_{#3}}
```

- There are a total three arguments. The first one (#1) is optional, and a default value has been supplied. The others (#2 and #3) are required.
- If a new value for the optional argument is **not** supplied, the default value of x will be used (in this case, in general it is whatever is in the second set of square brackets). Otherwise, the new input value will be used.

Example of a command with an optional argument

- Remember, the optional argument, if used, is in square brackets.
- Commands:

```
$\di \subvec{1}{n}$
```

```
$\di \subvec[a]{m}{m+k}$
```

- Outputs:

$x_1, \ldots, x_n$

$a_m, \ldots, a_{m+k}$

The Renewcommand Command

- The \renewcommand command allows us to redefine or alter an existing command.
- Here is a command that Dr. Fagerstrom uses when she runs out of alphabet on her review sheets:

```
\setcounter{enumi}{0}
\renewcommand{\labelenumi}{(\alph{enumi}\alph{enumi})}
```

- The first of these lines resets the counter to 0, and is something we could have done earlier in the lecture on list environments.
- The second of these lines 'overwrites' the command that defines the label, so that instead of (a), (b), etc., we get (aa), (bb), etc.
- **Be careful** when using \renewcommand. You can use it to accidentally overwrite standard LaTeX commands!