Math 291: Lecture 8

Dr. Fagerstrom

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

April 4, 2019

Table of Contents

Outline

1. Including .jpg Graphics
2. Figures
3. Labels and References
4. Theorem and Theorem-Like Environments
5. The amsthm package
6. Defining New Environments

Including .jpg Graphics Files

- To get things started, you will need to go to the course website and download some files.
- Go to the course webpage and download all of the picture files that were in the set of links for lecture 7.
- Put them in the folder that you are using to save your practice files and labs.

Including .jpg Graphics Files

- Use the package “graphicx”.
- **Note:** You need to compile using the profile: LaTeX => PDF
- The basic command to include a .jpg file is in the example below.
- Code:
  ```
  \includegraphics[width=1.5in]{marble4.jpg}
  ```
- You should see the following:

Including .jpg Graphics Files

- You should also note that [width=1.5in] is an optional argument that is being used to “size” the image.
- Experiment a bit to see how changing this number alters the output when you compile.
- An option for width is also [width=\textwidth]
- . . . or [width=0.5\textwidth] (since LaTeX knows how to multiply!)
- Another way to “size” an image file is to use the command [scale=#], where # is a decimal number that alters the size of the image.
- for example, [scale=.50] reduces the size of the image by 50%, and [scale=2.0] doubles the size of the image.

Including .jpg Graphics Files

- In addition to setting the width, you can also set the height.
- You can also rotate an included image specifying an angle as an optional argument, which sets an angle or rotation clockwise in degrees.
- The option: \keepaspectratio, which can equal either `true` or `false`, can be used to maintain the aspect ratio of the original image.
- And note that if you want it centered, you need to use the standard commands for centering in LaTeX.

Example

Code:

```
\includegraphics[width=4in,height=1in]{AttackMaybe.jpg}

\includegraphics[width=4in,height=1in,keepaspectratio]{AttackMaybe.jpg}

\includegraphics[height=1in,keepaspectratio,angle=-30]{AttackMaybe.jpg}
```

Example's output

Comments:

- Specifying both the height and the width can squash the image. The option keepaspectratio fixes that.
- When you specify all three (height, width, and keepaspectratio), LaTeX adjusts the image so that neither the width nor height specification is exceeded.
- In the angle command, note that you can use positive or negative angles, which are measured in the standard mathematical direction.

Outline

1. Including .jpg Graphics
2. Figures
3. Labels and References
4. Theorem and Theorem-Like Environments
5. The amsthm package
6. Defining New Environments

Figures

- Graphics and figures are different things to LaTeX.
  - Graphics are just pictures that are added, but are viewed as if they are just any other content within the document/paragraph.
  - Figures are separate environments within LaTeX. In particular, figures are automatically moved to where they fit, with certain restrictions, so are not always in exactly the same place in your document as you make edits to the document text. (Which is the way that pictures are often viewed in other word processors.)
- Figures can be added to documents, and the graphic is added separately within the figure environment. A graphic is just the most common type of figure, but is not required.
- Captions are allowed (and expected). You put the caption command below the graphic if you want the caption below the picture, and on top instead if you want the caption above the picture.

Figure Commands

- \begin{figure}[placement] . . . \end{figure}
- So figures are *environments*.
- The options for the placement parameter are h, t, b, and p (which stand for: here, top, bottom, and page).
- Figures are automatically numbered.
- To get a list of figures, usually at the start of a document after the table of contents, the command is
  \listoffigures
- The content of the generated list of figures consists of the captions of the figures.
- Note that, like the title page and the table of contents, the list of figures is typically on its own page. So you need to use the \newpage command.

Figure Commands

- See the 2019-291-ExampleDocWks7and8 documents to see examples on how the figures are added.
- Note that captions are usually placed below the graphics within the figures. But try moving one to just after the \begin{figure} command to see what it looks like when it is above the graphic.
- Note that the figure itself is page-wide and is automatically centered (as seen by the captions), but the graphic is not. The centering of the graphic is done within the figure environment. See what happens if you change the caption on Patch (figure 2) in the example file to just "The cat Patch.".
- Note that figures can, and often do, have multiple graphics, as in Marble's pictures (figure 5) in the example file.
- Note that the figure numbering is completely separate from the section numbering within the document.

Outline

1. Including .jpg Graphics

2. Figures

3. Labels and References

4. Theorem and Theorem-Like Environments

5. The amsthm package

6. Defining New Environments

Creating Labels

The syntax for labeling a numbered object is: \label{NameOfLabel}

- Labeling can be done anywhere within the body of the environment you are labeling but you are **strongly** encouraged to put the labels **immediately** after the command that creates the object you are labeling (such as the \section command).
- Labels must be placed at the appropriate "level".
- The name you give you label can be whatever you want it to be.
- It is common in long documents to use something like: \label{Fig:NameOfFigure} or \label{Chap:NameOfChapter}.
- Using this syntax, the names are just easier to remember.
- The name you use **never** appears in your final document.
- Figure labels must occur within the *caption* of the figure.

Referencing Labels

- When we reference a label, the compiler will insert the number associated with the object we previously labeled in the place where we type the reference command.
- Note that **only** the number will be inserted. We will usually add our own accompanying text to smooth out the phrasing of the reference. For example, we may want to use phrases in our code like:
  - "In Chapter `\ref{label}`,..."
  - "...back in Section `\ref{label}`, ..."
  - "...as we see in Figure `\ref{label}`..."
  - "...using Theorem `\ref{label}` ..."
  - "...as in part (`\ref{label}`) above ..."

Referencing Labels

- Specific labels are referenced using the command: `\ref{NameOfLabel}`.
- Note that you will need to run LaTeX **twice** in order for the labels to appear correctly.
- LaTeX will show *warnings* (**not** errors) when the labeling doesn't work.
- For this reason, when we compile, we may need to read the log file a bit more carefully when we are using labels and references.
- If a reference is bad, or if you forgot to run LaTeX a second time, ?? will appear in place of the label in your final document.

Referencing Labels

- Remember there is also the \pageref command (see the example document) that refers to the page number that the referenced object as well.
- In addition, there is the varioref package, which makes some adjustments to the verbage used in the referencing.
- See https://ctan.org/pkg/varioref?lang=en for more details.

Outline

1 Including .jpg Graphics

2 Figures

3 Labels and References

4 Theorem and Theorem-Like Environments

5 The amsthm package

6 Defining New Environments

Theorem-Like Environments

- There is a command in LaTeX that allows you to create environments that are appropriate for theorems, definitions, etc.
- Code: \newtheorem{Env name}{Title}[subctr]
- In this command, 'Env name' is the name used to call the environment.
- 'Title' is the title or name that is actually printed (along with with a "counter") when the document is compiled.
- 'subctr' is the counter of some other environment that this environment is embedded within.
    - If omitted, numbering is maintained throughout the document.
    - If present, it must be a standard LaTeX counter, and will create numbers such as '4.1', where 4 is the current value of the standard counter and 1 is the first use of the new environment (since the counter that generated the 4 was updated). Ex: Theorem 4.1 in section 4 of your document.

Theorem Example

- Code in the preamble of your document:

```
\newtheorem{thm}{Theorem}
\newtheorem{ax}{Axiom}
```

- Code in the body of the document:

```
\begin{thm}[The Fundamental Theorem of Calculus]
$$\int_{a}^{b} f(x)\, dx = F(b)-F(a)$$
\end{thm}
\begin{ax}
All dogs chase postal workers.
\end{ax}
\begin{ax}
All postal workers deliver mail.
\end{ax}
```

Theorem Example

Output should be:

Theorem 1 (The Fundamental Theorem of Calculus).

$$\int_a^b f(x)\,dx = F(b) - F(a)$$

Axiom 1.

All dogs chase postal workers.

Axiom 2.

All postal workers deliver mail.

Theorems, continued

- Notice that LaTeX keeps track of the numbering for you.
- When you add or remove theorems, the numbering throughout the document is automatically updated (if you build twice).
- Note that when you call a new theorem, you can give it an extra title on a one-by-one basis, as for the FTC in the exercise.
- It is possible to refer to the numbers, and those reference labels will be automatically updated as well.

Another example

- In the preamble: \newtheorem{thm2}{Theorem}[enumi]
- In the body of the document:

```
\begin{enumerate}
\item This is the first enumerated item.
\item This is the second enumerated item.
\item This is the third enumerated item.
\begin{thm2}
This is the first numbered theorem after item three.
\end{thm2}
\begin{thm2}
This is the second numbered theorem after item three.
\end{thm2}
\item Blah, blah
\begin{thm2}
This is yet another theorem statement.
\end{thm2}
\end{enumerate}
```

Another example

Output:

1. This is the first enumerated item.
2. This is the second enumerated item.
3. This is the third enumerated item.

Theorem 3.1.

This is the first numbered theorem after item three.

Theorem 3.2.

This is the second numbered theorem after item three.

4. Blah, blah

Theorem 4.1.

This is yet another theorem statement.

Theorems, continued

- Note that we referred to a specific enumeration level in our environment definition.
- Other counters can be used, like sections and chapters in larger documents.

Theorems, continued

- There is another version of this that can create sub-environments that continue the numbering of the main environment.
- The syntax is:
  `\newtheorem{Subenv name}[main env]{SubEnv-Title}`
- With this, you can create a sequence of theorems such as: Thm 1, Thm 2, Cor 3, Cor 4, Thm 5, Thm 6, . . .
- There are more options with the amsthm package.

Outline

1. Including .jpg Graphics
2. Figures
3. Labels and References
4. Theorem and Theorem-Like Environments
5. The amsthm package
6. Defining New Environments

Package: amsthm

If you don't want to take the time to define environments yourself, you can use the `amsthm` package.

- Along with the standard environments, this package defines a `newtheorem*` version, used for un-numbered theorems
- It also defines three environment styles:
    - plain (bold title, then italics in the body)
    - definition (bold title, then normal text in the body)
    - remark (italicized title, then normal text in the body)
- You activate a style with the \theoremstyle{style} command. It then remains that style until you re-define the style.
- You can also create your own styles with a \newtheoremstyle command.
- You can still manually define other theorem environments as if you didn't load the amsthm package.

Package: amsthm, continued

- The “amsthm” package also defines the `\swapnumbers` command (in preamble before any `\newtheorem` commands), which puts the numbers *before* the theorem (as in: **1 Theorem**).
- Finally, it defines a `proof` environment (`\begin{proof} \end{proof}`). This environment:
  - is unnumbered
  - it starts with *Proof*
  - it ends with: □.

Outline

1. Including .jpg Graphics
2. Figures
3. Labels and References
4. Theorem and Theorem-Like Environments
5. The amsthm package
6. Defining New Environments

New Environments

- Finally, the following syntax can be used to define a new LaTeX environment:

  ```
  \newenvironment{envname}[narg][opt]{begdef}{enddef}
  ```

- ‘begdef’ is the stuff that is printed when the environment is opened, and ‘enddef’ is the stuff that is printed when the environment is closed.
- We can also define and make use of new counters using the command `newcounters`.
- These options are part of what makes LaTeX highly customizable and useful.
- For details on how to use the \theroemstyle, \newenvironment, and related commands, the internet is your friend! (In other words, Google it!)