Tips and Tricks

Math 351

This final week of Math 351 is for final design tips and for introducing some useful packages and commands we haven't yet encountered. Thank you for your hard work this quarter, I hope you've gotten something out of our course.

1 Design tips

1. Read the package or class documentation! The answers to typesetting questions are in the documentation.

   Our textbook and the example `.tex` files from Math 351 are other places to find quality typesetting advice. Warning: sometimes the answers on random websites and message board site such as stackexchange can be outdated and incorrect.

2. Do not copy or paste long, mysterious preambles from other `.tex` documents without understanding the commands (this goes along with reading the documentation).

3. Before defining an newcommand or environment, check CTAN to see if someone has already done the work. For example, don't come up with your own way to denote modular operations; for that there is the predefined commands `\pmod` and `\bmod` which can be used in this way: $a = b \pmod{c}$ and $a \bmod c = b$

4. Write the document first and make visual formatting choices later. Focus on content and do not fiddle with formatting too much while writing.

5. Spacing commands such as `\\`, `\vspace`, `\newpage`, or `\par` are rarely needed.

6. Graphics need to be of high quality. Never use pixelated or low resolution images.

7. If there is a long unbreakable word (see the `\mbox` command) or a long mathematics equation, LaTeX might place part of the long expression in the margins. If this happens, the compiler will register an "`Overfull \hbox`" error. Avoid this, usually by using `\gather`, `\multiline`, or `\align` for long mathematical equations.

8. Use `\emph` to *emphasize* text instead of using `\textbf` or underlining.

2 Tricks and sometimes useful commands

1. The `minipage` environment creates a smaller page within a page, useful for side by side type:

This is left text.

```
\begin{minipage}{15eM}
This is left text.
\end{minipage}
\begin{minipage}{15eM}
This is right text.
\end{minipage}
```

2. The `figure` and `table` environments create floating figures or tables (floating means LaTeX decides the best placement of the figure). The options `h`, `t`, or `b` tell LaTeX to attempt placing the figure here, top, or bottom of the page, respectively. To have words wrap around the figures, use the `wrapfig` package.

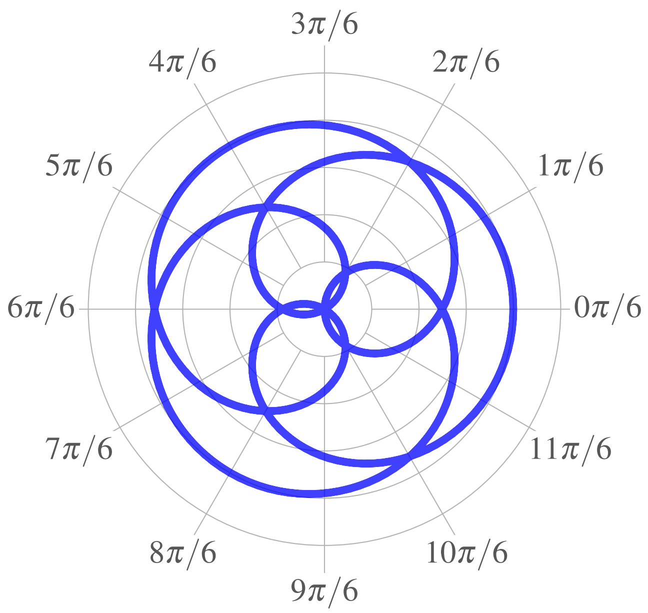

Fig. 1: The polar equation $r = \cos(3x/7)$ provides an example of a floating figure.

3. The compiler makes typesetting decisions by placing boxes around letters, parts of words, mathematics, and figures, and then appropriately arranging the boxes. Force unbreakable type to reside in a single box using `\mbox{unbreakable text}`. Create a frame around a box using `\fbox{.}` and an unbreakable paragraph using `\parbox{width}{.}`.

   Sometimes when using commands such as `\hfill`, an empty box is needed to get spacing just right; create such an empty box with `\mbox{}`.

   On a similar note, ~ is a non-breaking space character, used when a space between two words or characters should appear but those words cannot be on different lines.

4. Separate multiple authors using `\and` within the `\author` command.

5. Double spacing is achieved by placing `\linespread{1.6}` in the preamble.

6. Most of us are familiar with tabs from our frequent use of physical typewriters. Tabs can be kept and used with the `tabbing` environment. Set tabs using `\=`, create a new line using `\\`, and move to the next tab using `\>`. For example,

   The first tab appears right here, this is text after the first tab, and there is a third tab.
   This is the second row,   middle of second row,   and the end.

   is created using

   ```
   \begin{tabbing}
   The first tab appears right here, \=
   this is text after the first tab, \=
   and there is a third tab. \\
   This is the second row, \>
   middle of second row, \>
   and the end.
   \end{tabbing}
   ```

   Tabbing environments are treated as one box and thus cannot be split across two pages.

7. Custom counters can be defined using the syntax

   ```
   \newcounter{countername}
   \setcounter{countername}{1}
   ```

   in the preamble. The current value of the counter can be accessed using `\thecountername` and incremented using `\addtocounter{countername}{1}`. Counters work well in tandem with newcommands or newenvironments.