Slide 1

Administrivia

- Homework 7 (LaTeX) coming soon. Tomorrow? or Monday, when the plan is to do one more lecture on the topic.

Slide 2

What are TeX and LaTeX?

- TeX — program for typesetting mathematics, developed by Knuth (1978) for his book *The Art of Computer Programming* and made freely available. (How it came to be — a "side project" that turned into much more!)
- LaTeX — extensive set of macros for TeX originally written by Lamport (1985), that provide functionality needed for scholarly papers. Extended over many years by many people.
- These are "text formatters" not "word processors", and as such don't include a built-in editor. (But there are IDE-like programs for working with them.)
- Basic idea — you write "source code" for your document (text and markup) with a text editor, then use TeX or LaTeX to turn it into a formatted document.
- Both available in zero-cost form for many platforms.

Slide 3

Basics (Under UNIX)

- You write "source" (`foo.tex`) with a text editor of your choice. It includes your text plus "logical markup" — e.g., `\section{A Section Heading}`.

  (What about checking spelling? Use a separate tool — "each program should do one thing, and do it well." `ispell` and `aspell` are common ones.)
- Traditionally, you use the command `latex` to generate a `.dvi` file, then `dvips` to generate PostScript, then (if desired) convert to PDF with `ps2pdf`. You can also go directly to PDF with `pdflatex`.

Slide 4

Isn't That a Lot of Trouble?

- In some ways, yes — there is a learning curve, and there are many "gotchas".
- For some jobs (where visual layout matters more than logical structure), LaTeX is probably the wrong tool.
- But if you persevere . . .

Slide 5

Why It Might Be Worth the Trouble

- Output looks good — math in particular.
- Logical structure of document is clearly spelled out. (You can sort of do this with, e.g., MS Word, but it's less transparent.)
- Cross-referencing, bibliographic references, footnotes, tables of contents, indexing, etc., "just works".
- Documents are stable — only way to "corrupt" a document is to mess up with your text editor. Very old documents usually still compile, and if they don't the content is still accessible.
- Once you figure out how to do a particular trick, it's there in the `.tex` source for future reference.
- If you want to generate a formatted document programmatically, LaTeX source may be a good target.

Slide 6

Basics, Continued

- LaTeX provides a small set of "document classes" — article, report, book, etc. These classes group definitions for section headers, lists, etc., in a way that everything looks good together. Also can have "packages" that group together related customizations, provide extra features.
- Basic document structure (look at example):
  - `\documentclass[options]{foo}`
  - Additional global definitions, packages, etc.
  - `\begin{document}`
  - Your text. "Paragraphs" delimited by blank lines.
  - `\end{document}`

Slide 7

Some Features

- "Sectioning commands" provide consistent layout and automatic numbering. Also can be used to generate table of contents.
- "Environments" provide support for lists, tables, centered text, "verbatim", etc.
- Predefined macros provide simple markup, e.g., `\textit{foo}`.
- Math — a bit cryptic, but IMO not worse than point-and-click equation editor.
- Graphics can be included. Some details next time.

Slide 8

Minute Essay

- What do you currently use to produce formatted documents? What do you like/dislike about it?
- Have you tried LaTeX? If so, what do/did you like/dislike about it? Anything you'd like to know how to do but don't?