Introduction to Scientific Typesetting
Lesson 15: Software to Help with $\LaTeX$

Ryan Higginbottom

January 25, 2012
LaTable
LaTeXDraw
An Overview

LaTable
The Problem of Spreadsheets
The Options
LaTable
Strengths and Weaknesses of LaTable
Practice!

LaTeXDraw
There are situations you can imagine where some data that you want in a document or presentation is contained in a spreadsheet.

But—in terms of how the text is entered—the *tabular* or *array* environments are much different than spreadsheets!

We need a way to get spreadsheet stuff into \LaTeX.
There are two main options of which I’m aware.

1. **The *Excel2LaTeX* Macro for Microsoft Excel.**
   
   Highlight a range of cells in an Excel spreadsheet, click a button, and get **LATEX** code.
   
   - **Positive:** does nice job with text decorations (bold, italics)
   - **Negative:** some inconsistent behavior with cell borders and cell justifications, can only generate code for **tabular**

   Check out the `spreadsheet example-1.xls` and `example15-1.tex` files.

2. **The *LaTable* program.**
LaTable reads **comma separated value** (`.csv`) files.

Convert Excel file to `.csv` format, then open in LaTable.

Let’s practice with `spreadsheet example-1.xls`.
Strengths and Weaknesses of LaTable

- **Strengths**
  - Can edit cells so that code comes out cleaner
  - Does a great job with cell justification and cell borders
  - Merged cells (across columns) also show up well
  - Can export to \LaTeX\ code as \texttt{tabular}, \texttt{array}, or any custom environment

- **Weaknesses**
  - Cannot do text decorations
  - Cannot merge cells across rows
Take the `spreadsheet example-2.xls` file and, using LaTable, reproduce the second example file (.pdf).
LaTeXDraw

An Overview

LaTeXDraw

Difficulty with pstricks
Using LaTeXDraw
First Example
Including Drawings as Images
Lots of Options
Practice!
Why Use Anything Else?
Let’s face it—sometimes \texttt{pstricks} can be pretty difficult to use.

There are several programs which allow you to draw and generate \texttt{pstricks} code which you can paste into \LaTeX.

\LaTeXDraw is one such program.
This is a fairly amazing program, which allows you to draw all types of shapes.

- squares and rectangles
- circles and ellipses
- dots, lines, polygons
- arcs and curves
- grids and axes
- text
- free-hand drawings!
Open LaTeXDraw and then open the `drawing.svg` file.

We will take the generated `pstricks` code and build a `.pdf` file from this.

You’ll need to add a few lines of code to make a `.tex` file that will build:

- `\documentclass{article}
- `\begin{document}` and `\end{document}`
- Uncomment the four `\usepackage{...}` lines.

By the nature of the generated code, these drawings are really easy to `scale`. Just change the number in the `\scalebox{#}` line.
LaTeXDraw has a really great feature that allows you to use it with the LaTeX => PDF build profile. (Remember: pstricks requires the LaTeX => PS => PDF profile.)

Draw something and click on the Adobe button toward the upper left of the menu bar. You’ll get a PDF file which is just as big as the pictures you’ve drawn.

Now include this in a document using the graphicx package and the \includegraphics command.

**An alternative:** under the “File--Export as...” menu option the picture can be exported as a .jpg or any other image file format. This also allows inclusion in a document using the LaTeX => PDF build profile.
Let’s look closely at a few of the many things to customize when drawing with \LaTeXDraw.

- The colors associated with a geometric object.
- The many options associated with a text box.
- The many options associated with axes/grids.
Open the third example file (.pdf).

Use LaTeXDraw and include \texttt{pstricks} code into a \texttt{.tex} document to reproduce it.
In my view, this is a nice quick way to do some things—particularly free-hand drawing.

But for things like function graphs and node/node connections, writing the code by hand is still the way to go.