

How to give a L^AT_EX supported Presentation

Jason Miller

23 April 2002

This document is a “paper” version of a slide presentation. By writing this, I intend to help you create a slide presentation for your Capstone that looks like a PowerPoint presentation but is better insofar as it uses no Microsoft products and displays mathematical symbols in a respectable manner.

I will presuppose that you have a L^AT_EX document that contains the information that you want to give in your presentation. Let’s imagine that you’ve done this in the article documentclass and that you’ve separated slide in the logical way: by using L^AT_EX’s `pagebreak` command

The document won’t look like a slide show, but at least it’ll have your material in it.

An examples of such a file would be `PreSlideFile.tex` which you can find under the “LaTeX does PowerPoint” link on my WWW page: <http://vh216801.truman.edu/millerj>.

1 pdf_lat_ex

You will use Adobe’s Acrobat Reader to give your presentation. For this reason, your L^AT_EX output cannot be in DVI format; it needs to be in Portable Document Format (PDF). The absolutely best way to get your output into this format is to use the L^AT_EX compiler `pdflatex`. This compiler is very similar to the standard L^AT_EX compiler with a couple exceptions. `Pdflatex` outputs PDF instead of DVI. It barfs when you have graphics files in EPS format; instead it would like graphics in PDF format. The good news is `pdflatex` can work with graphics files in JPEG, TIFF, and PNG format graphics.

More on this later.

2 Foils

The first thing you want to do to your pre-slide document is to change `article` in your documentclass declaration to `foils`. In fact, as long as you’re at it, you might as well replace that declaration with the following

```
\documentclass[landscape]{foils}
\usepackage{fancyhdr,amsfonts,amsmath,amsthm}
\usepackage[pdftex]{color}
\usepackage[pdftex]{graphicx}
\usepackage{times}
\usepackage[pdftex]{hyperref}
```

This means that you'll require some unusual style files (*e.g.* foils) but it'll be worth it. you can find these style files at the same WWW location as `PreSlideFile.tex`.

If you now compile your \LaTeX source file you'll see that it actually looks like you have a collection of slides that you could almost use for your talk. In fact, you could print these pages out on a laser printer and use a document camera to practice your talk *right now!* If you don't have a \LaTeX file of your own that you're working on as you read this, take a look at `PreSlideFile-foil_v1.tex` and compare its output to that of `PreSlideFile.tex`.

But don't go do that quite yet. Instead, do a search-and-replace on your \LaTeX source to replace all instances of `\pagebreak` with the `\foilhead{}` command. You'll see that this makes some subtle changes in your document (mostly relating to spacing).

At this point you haven't used the optional argument in the `\foilhead{}` command. By putting text or mathematics in between the braces we've left empty you can create a title line for a slide. I've done this for the first slide after the title slide. I put the words "Category Theorey" in the foilhead command. Notice what it does.

3 Color

At this point you can make perfectly good slides for giving an electronic, \LaTeX based presentation. Just print out the slides on your nearest printer and you're good to go.

If you have a color printer, you might want to add color to your slides. To do so, use the `\textcolor` command. For example,

```
\textcolor{red}{whazzah!!}
```

produces **whazzah!!**. You could exchange the word **green** or **blue** for **red** and get **whazzah!!**. or **whazzah!!**, respectively. Other colors that are predefined include **cyan** **magenta** and **yellow**. You can define your own colors using the `\definecolor` command. To learn how this is done, talk to your favorite local \LaTeX afficianado.

4 Electronic presentation

If you plan to give your presentation using a computer, add the following command to your preamble:

```
\hypersetup{
  pdfpagemode={FullScreen},
}
```

This `hyperref` command (note that if you have done as I directed above, you're \LaTeX file will load the `hyperref` package) will embed a command in your PDF output that will direct Acrobat Reader to open your presentation in full-screen mode.

I've added this to the sample file `PreSlideFile-foil_v3.tex`.

Note that when your presentation is in Acrobat's full screen mode, the escape key will put it back into "normal" viewing mode. The up- and down-arrows advance and reverse the slides.

5 PPower4

Now, if you're not satisfied with the look and feel of a `foils`-based \LaTeX presentation, and you *really* want that PowerPoint-like appearance, I have some good and bad news.

The good news: it's possible.

The bad news: it requires simple post-processing using a java program and the Java Virtual Machine.

The post-processor is called PPower4, and is freely available for download somewhere on the internet. (See the PDF presentation called `presentations.pdf` for this information.) When you install this, the post-processing is simple. First, you should add the following to your source document's preamble:

```
\usepackage{pause}                %required for ppower4
\usepackage{background}           %required for ppower4
\usepackage{pp4slide}             %required for ppower4
```

These will allow you to give your presentation that basic PowerPoint look and feel.

As long as you have these packages loading when you compile, let's use them. The `pause` package allows you to use the `\pause` command in your \LaTeX file. This command pauses the drawing of a page at your mark. In `PreSlideFile-foil_v4.tex` I've put some `\pause` commands on the second slide so the items in my `itemize` environment appear one-at-a-time.

Ok, now I need to point something out so you don't get freaked out. If you `pdflatex` your \LaTeX source now and look at the PDF file you obtain, you should see that there's not much text visible and that there are little orange boxes where you think you put your pause commands. This is good. This is what you should see. Once you run your PDF file through PPower4, everything will look perfect. Let's do that now.

I invoke java from the command line, so I use PPower4 in the following manner. At the command line I type:

```
java -jar pp4p.jar PreSlideFile-foil_v4.pdf tmp.pdf
```

Here `pp4p.jar` is the Java program that does the post-processing, `PreSlideFile-foil_v4.pdf` is the PDF file created by `pdflatex` (the one with not much text and the orange boxes), and `tmp.pdf` is the PDF presentation I'll use when I give my talk. Take a look at it. Open it using Acrobat Reader.

According to my computer, I have a set of slides with white text on a solid PowerPoint-blue background. By bullet points are red dots and my slide titles (from the `\foilhead` command) are bold and yellow-ish. And the pauses on the second slide are cool!

6 PPower4 Options

Put the following lines in your preamble:

```
\definecolor{DarkGreen}{rgb}{0.5,0.8,0.6}    %define a custom color
\definecolor{RGBblack}{rgb}{0.0,0.0,0.0}     %define a custom color
\pagecolor[blue]{RGBblack}
```

They define two new colors and use them to give all your slides a background color of a blue-black gradient. This is very PowerPoint-like.

You can repeat the `\vpagecolor` command on any of your pages to change the background color of your slides. There are other variations on this command. See the `presentation.pdf` example for more info.)

There are more goodies that you can invoke using PPower4. My understanding is that PPower4 lets you pass PDF commands to Acrobat, so you can make your presentation do anything that an Acrobat PDF file can do. How to do those things is, right now, a mystery to me. When I learn more, I'll be happy to pass my knowledge on to you.

7 Summary

To make a \LaTeX based slide presentation (paper or electronic), do the following:

- Create a basic \LaTeX document using the `article` style that contains the text you would like to put onto your slides. Separate slides by using the `\pagebreak` command.
- Change your documentclass declaration from `article` to `foils` (and declare the `landscape` option instead of a point size option). This will make your document look more like slides.
- Replace the `\pagebreak` commands with the `\foilhead{}` command. This will modify each slide slightly, and if you put text or mathematics into the argument of the `\foilhead{}` command, the text or math will appear as a title on the foil...on the slide.
- Adjust spacing and colors as needed and you have a perfectly good set of slides to use for giving a presentation.
- Add color by using the `color` package and its `\textcolor{}{}` command.
- Use `pdflatex` to create PDF output that can be used in conjunction with Adobe Acrobat reader to give a computer-based presentation (think: PowerPoint).
- If you have access to a computer that has a Java Virtual Machine installed, you can post-process your \LaTeX source (after making a few additions to the source file) to produce a slick looking, PowerPoint slammin', \LaTeX based presentation that will wow your friends and enemies.

(I think) all the files referred to in this document are available at <http://vh216801.truman.edu/miller>
Happy presenting.