Graphics examples in LaTeX

John Kerl • http://johnkerl.org

April 26, 2010

Contents

1 Creating figures

For plots, I sometimes use Matlab. (See figure 1.) Go into Matlab, do your plot, set up axis labels as you wish, etc. Then save as EPS format.

More often, for plots I use Python's `pylab` module, which has Matlab-like plotting functionality. I also use my `pgr` script, which is a handy wrapper around `pylab`: it is a script which reads tabular file data and command-line arguments, then calls the desired `pylab` functions. Please see `http://johnkerl.org/python/doc/pgr.html` for more information about `pgr`. See figure 2 for a plot.

For drawings, I use Inkscape: figure 3. The difference between the two sides of the figure (i.e. whether or not you see TeX fonts in the figure or not) will be discussed in section 3.

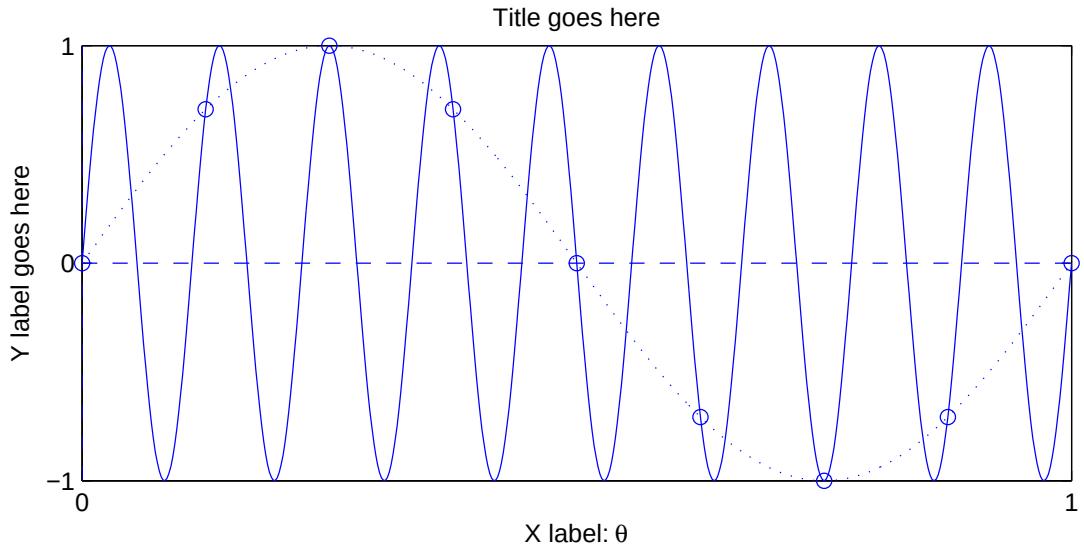

Figure 1: Caption for Matlab figure goes here.

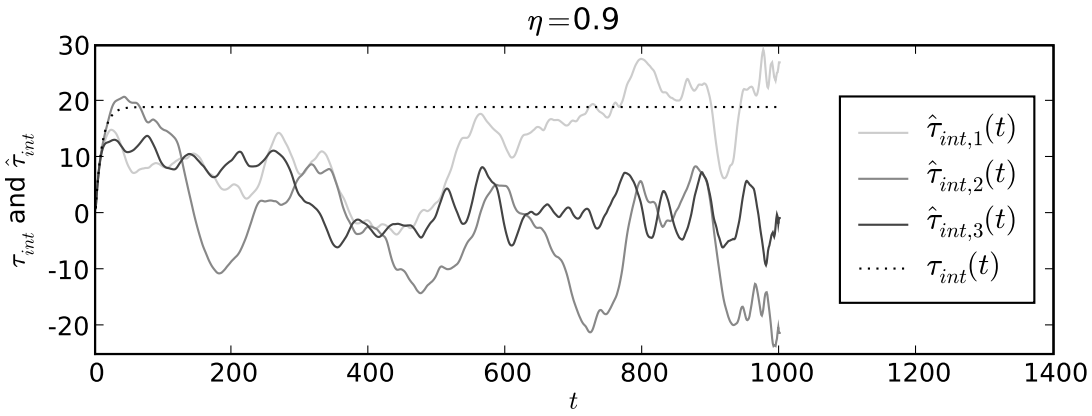

Figure 2: Caption for `pylab` plot goes here.

2 Including figures in your document

First, in the preamble (i.e. above `\begin{document}`), put the following:

```
\usepackage{graphicx}
\usepackage{psfrag}
```

Then, put the following in the body of your document:

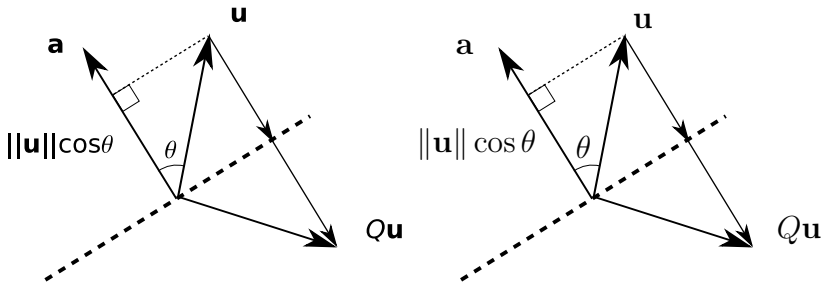

Figure 3: Caption goes here.

```
\begin{figure}[!htb]
\begin{center}
\psfragscanon
\includegraphics[scale=0.8]{figures/tauint_eta_0.9.eps}
\caption[Shorter caption for the list-of-figures page (if you have one).]
    {Caption for \texttt{pylab} plot goes here.
    \label{fig:pgr}}
\end{center}
\end{figure}
```

3 Putting mathematical text in figures

There are (at least) two options: (1) use the plotting/drawing program's fonts for the figure; (2) use TEX fonts for the figure.

For Matlab: If you type in things like `\theta`, Matlab will convert it to a symbol for you when it creates the EPS file. See also `help text` in Matlab.

For `pylab`: here also, most everything like `$\theta$` will be converted to a symbol for you when `pylab` creates the EPS file. Make sure to include dollar signs around any TEX-style expressions.

For Inkscape, or any drawing program which doesn't convert TEX-style symbols for you, you can use the program's own fonts for the figure. In my experience, drawing programs usually have very limited (or awkward-to-use) support for mathematical symbols. To get TEX fonts for the figure, you can do the following. Instead of trying to get Inkscape to create a θ symbol, just put the text

```
\tex{$\theta$}
```

into a text box in Inkscape. Then save your SVG file. Then save as EPS also, but be sure

to uncheck the *Convert text to paths* option. Here's where the LaTeX package `psfrag` comes in. When you run your `.tex` file through the `latex` command, it interprets the symbols in your figure's EPS file. Voilà — now the symbols in your figure match the symbols in the body of your paper.

The disadvantages of using `psfrag` are (1) The DVI previewer `xdvi` won't interpret the `\tex{$...$}` strings correctly; instead, you can view the PDF version of your document to see the strings. (2) You no longer have WYSIWYG (what you see is what you get) inside the drawing program. Instead, you may need to repeatedly iterate: save the figure file as SVG, save as EPS, close the PDF viewer, build the PDF file for your paper, open the PDF viewer and look at the figure; then go back into the drawing program and moves the math symbols around ....

To summarize:

- In the drawing program, use strings of the form `\tex{$...$}`. Don't bold these strings in the drawing program, change the font size, etc.
- In the drawing program, save the figure as EPS.
- In your LaTeX file, have

  ```
  \usepackage{psfrag}
  ```

  in the preamble. Also have the `psfragscanon` line for each figure, as shown:

  ```
  \begin{figure}[!htb]
  \begin{center}
  \psfragscanon
  \includegraphics[scale=0.8]{figures/tauint_eta_0.9.eps}
  \caption[Shorter caption for the list-of-figures page.]
      {Caption for \texttt{pylab} plot goes here.
      \label{fig:pgr}}
  \end{center}
  \end{figure}
  ```

- Generate a PDF file from your LaTeX file as shown next.

4 Generating DVI and PDF files

This file (and the directory containing it) serve as a copyable example. Files:

- `grex.tex`: The LaTeXsource for what you're reading now.

- `isomath.tex`, `mathenv.tex`, `syms.tex`: misc. environment and symbol definitions.
- `figures/inkfig.svg`: An Inkscape file, in SVG (scalable vector graphics) format. Just run Inkscape (`inkscape &` at the Unix prompt on the math department Linux machines) and draw. Inkscape has nice on-line help. Then save.
- `figures/inkfig.eps`: In Inkscape, simply do save as EPS. Then accept the defaults when prompted: *Make bounding box around full page* = false, *Convert texts to paths* = false, *Embed fonts (Type 1 only)* = false.
- `figures/matlab_figure.m`: A Matlab script file. Edit this file to see how it works; consult Matlab documentation for more information. Run Matlab (`matlab &` or `matlab -nodisplay` on the math department Linux machines), then type the name of the file without the `.m`. Note that the file `figures/matlab_figure.m` contains the name `matlab_figure.eps` inside it, so if you rename the `.m` file, also don't forget to edit and change the last line.
- `figures/tauint_eta_0.9.txt` and `figures/mktauintplot.sh`: The former is a data file. The latter is a shell script which invokes `pgr` with the command-line options necessary to produce figure 2.
- `dbuild`: A shell script to make a DVI file. For this file, it simply contains the line `latex grex.tex`.
- `pbuild`: A shell script to make the PDF file you're reading. I use `latex`, `dvips`, and `ps2pdf` to create the sequence `.tex` → `.dvi` → `.ps` → `.pdf`. You can do this all in one step using `pdflatex`, but `pdflatex` won't read EPS files. So, my `pbuild` script contains the following commands:

```
dbuild
dvips -o grex.ps grex.dvi
ps2pdf grex.ps
rm grex.ps
```