$\mathcal{ANUMS}$ LaTeX Workshop II

Maxim Jeffs

July 14, 2016

This session outlines a number of useful tools for working with LaTeX – feel free to skip to those sections that interest you. The sections primarily give brief introductions to the possibilities along with references to resources for learning more, rather than attempting to be comprehensive. The PDF file of this sheet as well as the LaTeX code used to create it are both available at `MaximJeffs.github.io`.

1 Equation alignment with `amsmath`

Typesetting long equations or equations broken across lines is somewhat problematic. The package `breqn` *attempts* to do this automatically, but does not usually succeed, so it is typically necessary to do this manually using the `amsmath` package. If you wish to typeset a long equation or a collection of several equations, begin with the commands `\begin{align}` and `\end{align}`. Adding mathematics inside the `align` environment works just as with adding entries to a matrix: use `&` to change column and `\\` to change row. For instance

```
\begin{align}
x  &= y  + z  &  j  & = k \\
   &= p  + q & \ell & = m\\
\end{align}
```

will produce

$$\begin{aligned} x &= y + z & j &= k \\ &= p + q & \ell &= m \end{aligned}$$

For much more on equation alignment see:

- The page at `www.sharelatex.com/learn/Aligning_equations_with_amsmath`, which compares multiple different methods for alignment;
- The outline in `en.wikibooks.org/wiki/LaTeX/Advanced_Mathematics`, which also includes many other important tools;
- Or the extremely comprehensive `ctan.unsw.edu.au/info/math/voss/mathmode/Mathmode.pdf`.

Note that lists such as the above can be created using `\begin{itemize} \item text \end{itemize}`. The hyperlinking was done using the package `hyperref`

2 Figures with `graphicx`

Inserting figures into LaTeX documents is fairly easy using `graphicx`; begin by adding `\usepackage{graphicx}` to the start of the document, and the image, as a PDF file to the same folder as your `.tex` document, with file name, for instance `example.pdf`. The commands

```
\begin{figure}
\centering
\includegraphics[width=0.6\textwidth]{example.pdf}
\caption{An example figure, courtesy of Wikipedia.}
\label{example}
\end{figure}
```

will then produce Figure 1 below. Note that the `width` of the figure can be set to any multiple of the width of the text, simply by changing the number in front of `\textwidth`. The `\label` command allows you to refer to the figure later in the document by typing "See `\ref{example}`", to produce: See Figure 1. This can be very helpful when you have a large number of figures whose order is likely to change during the course of writing.

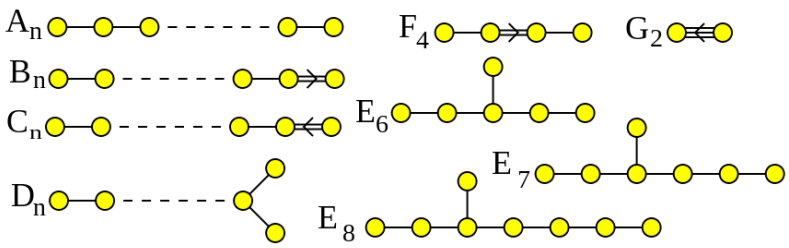


Figure 1: An example figure, courtesy of Wikipedia.

So long as you choose to compile with `PDFLaTeX`, your image file can be in any of the common image formats, such as `PNG`, `JPG` or `PDF`. However, it is recommended that you only include graphics that are in a vector format, such as `PDF`. The program InkScape is very useful for creating PDF mathematical figures and can be downloaded from `inkscape.org/en/download`. The pages at `inkscape.org/en/learn` may also be helpful.

One point to be wary of is that LaTeX will decide for itself where to put your figure; although it is usually fairly good, it may end up putting your figure several pages after you reference it. If this is the case, try placing `[h]` after `\begin{figure}`, and this will force the figure to be placed as close as possible to the location in the text. It should be noted however that this is not always considered to be best practice; if this really concerns you, see the extremely long answer at `tex.stackexchange.com/questions/39017`.

3 Macros with `newcommand`

Adding new commands can make typesetting mathematical documents much faster. To create a simple new command, add `\newcommand{name}{what you want it to do}` to the beginning of your document, after the `\usepackage` commands. You can then use `\name` to insert the command you have written. For example

```
\newcommand{\glnr}{ $\mathrm{GL}_{n}(\mathbb{R})$ }
```

allows one to type `\glnr` in text and produce $\mathrm{GL}_n(\mathbb{R})$. It is also possible to create new commands with *arguments*, using the code `\newcommand{name}[n]{thing involving n inputs, referenced by #i}`. For instance, writing

```
\newcommand{\pd}[2]{\frac{\partial #1}{\partial #2}}
```

allows one to type `\pd{x}{y}` inside an equation and produce $\frac{\partial x}{\partial y}$. One can also have more sophisticated examples:

```
\newcommand{\matrix}[1]{ \left( \begin{matrix} #1 \end{matrix} \right)}
```

which will create a matrix inside an equation, into which you can enter entires as usual. One other useful feature is `\let`, which allows you to change the default commands. For instance

```
\let\varepsilon\epsilon
\let\epsilon\varepsilon
```

will change `\epsilon` to produce ε instead of ϵ. Finally, `\DeclareMathOperator` allows one to define 'operators', such as

```
\DeclareMathOperator*{\Res}{Res}
```

so that

```
\Res_{z = 1} \sum_{n=1}^{\infty} \frac{1}{n^{z}} = 1
```

produces

$$\operatorname*{Res}_{z=1} \sum_{n=1}^{\infty} \frac{1}{n^z} = 1$$

Other useful types of new commands may can be found at `en.wikibooks.org/wiki/LaTeX/Macros`, such as `\newenvironment` and so forth; you can even write new commands that perform arithmetic operations on their arguments, as well as conditionals and loops!

4 Theorems with `amsthm`

The Definition-Theorem-Proof format of mathematical papers is easy to achieve in LaTeX. Begin by adding `\usepackage{amsthm}` at the beginning of your document, as well as the following incantation

```
\newtheorem{theorem}{Theorem}
\newtheorem{definition}{Definition}
```

Now automatically numbered Theorems and Definitions an be created by typing into a `\begin{theorem}` environment, and proofs with `\begin{proof} \end{proof}`. For instance,

```
\begin{definition}
A group is said to be \textbf{simple} if it has no proper non-trivial normal subgroups.
\end{definition}
\begin{theorem}
The only finite non-trivial abelian simple groups are of prime order.
\end{theorem}
\begin{proof}
Trivial.
\end{proof}
```

produces

Definition 1. *A group is said to be* ***simple*** *if it has no proper non-trivial normal subgroups.*

Theorem 1. *The only finite non-trivial abelian simple groups are of prime order.*

Proof. Trivial. □

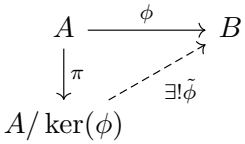
Lemmas, Propositions, Corollaries, Remarks etc. may be added in the same fashion; referencing theorems may also be done with the `\label` and `\ref` commands described in Section 2. For more on the possibilities of `amsthm`, including customisation of styles and numbering, see for instance this page.

5 Diagrams with `tikz-cd`

The package `tikz` is a general drawing package with wide capabilities; `tikz-cd` is a variant of `tikz` that is particularly useful for drawing commutative diagrams. The language works in a similar way to that for matrices except that you may produce arrows using the command `\arrow{direction}{name}`. This is perhaps best illustrated through a simple example:

```
\begin{center}
\begin{tikzcd}
A  \arrow{r}{\phi} \arrow{d}{\pi}  & B \\
A/\ker(\phi) \arrow[dashed]{ru}[swap]{\exists! \tilde{\phi}}
\end{tikzcd}
\end{center}
```

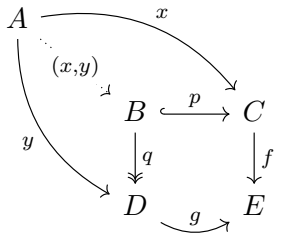
will produce



The best part about `tikz-cd` is the ease with which it allows you to produce all manner of strange arrows:

```
\begin{center}
\begin{tikzcd}
A\arrow[bend left]{drr}{x}
\arrow[bend right]{ddr}[swap]{y}
\arrow[dotted]{dr}[description]{(x,y)} & & \\
& B \arrow[hook]{r}{p} \arrow[two heads]{d}{q} & C \arrow{d}{f} \\
& D \arrow[bend right]{r}{g} & E
\end{tikzcd}
\end{center}
```

produces



The introduction at `www.jmilne.org/not/Mtikz.pdf` is very helpful and contains a wealth of examples, as well as a nice comparison of the different packages for drawing commutative diagrams. The general `tikz` language is more complicated but also much more powerful: see for instance:

`cremeronline.com/LaTeX/minimaltikz.pdf`

for an excellent short introduction. If you are not planning on creating anything terribly complicated, `TikzEdt` at `www.tikzedt.org` allows you to draw a diagram and convert it into Tikz code. Unfortunately, there is currently no version for Mac, and the user must be warned that it is quite unpolished. The website `www.texample.net/tikz/examples` is also extremely helpful if you can find a similar diagram and modify it. You are also warned in advance that `tikz-cd` does not interact well with the `proof` environment.

6 Referencing with `bibtex`

To use BibTeX you will need to create a text file ending with `.bib` in the same folder as the `.tex` file for your document, into which you will type your bibliography. The `.bib` file contains a series of 'entries', prefaced with the type of source, with the first line containing a collection of letters that you will use to refer to it, followed by the bibliographic information. For example, the `.bib` file might contain

```
@book{somelabel,
       author = "Kobayashi, S. and Nomizu, K.",
       title = "Foundations of Generalised {D}irac Geometry",
       volume = "I",
       publisher = "Wiley Interscience",
       year = "1963" }
```

This file can be created simply using a normal text editor. For each type of document, you will need a different BibTeX template where the information can be entered, such as `@book` or `@article`: a useful list can be found on the page `www.verbosus.com/bibtex-style-examples.html`. However, writing BibTeX entries by hand is a complete pain: there are multiple sites that will do it for you. The site `truben.no/latex/bibtex` will let you to enter the bibliographic information and convert it to a BibTeX entry, and MathSciNet at `www.ams.org/mathscinet/index` allows you to search for papers and copy the bibliographic information already in BibTeX format. Alternatively, if you are using a Mac, there is also the software `BibDesk`. One important thing to note however is that proper names in titles in a BibTeX entry will not be capitalised in the bibliography unless they are placed in brackets (see `{D}irac` above) and that many of these tools will not do this automatically.

Once you have written the bibliography, enter the incantation

```
\bibliography{biblio}{}
\bibliographystyle{amsalpha}
```

wherever in the document you want the bibliography to appear. If you would rather have numbered citations, change `amsalpha` to `plain` above. Now, to reference a bibliography entry in text, simply write `\cite{somelabel}` or `\cite[p.~81]{somelabel}`, and this will produce, for instance [KN63] or [KN63, p. 81] respectively, as well as the **References** section below. In order to get the bibliography entries and citations to show up in the PDF for the *first time*, you must compile the file first with LaTeX, then with `bibtex`, then with LaTeX *two more times*. How you change the compilation settings will vary between editors: if you are using TeXShop or TeXWorks, there should be a small selection box at the top of the screen that says LaTeX; scroll down until you find BibTeX and hit `Typeset`, as usual. Unless you subsequently edit the bibliography, you can compile as usual, but otherwise you will have to complete the same process again. If you ever see small boxes like `[?]`, this means that you need to compile with LaTeX again. Some LaTeX editors (such as TeXMaker) allow you adjust settings so that all four steps happen automatically every time you compile; see the specific documentation for details.

References

[KN63] S. Kobayashi and K. Nomizu, *Foundations of generalised Dirac geometry*, vol. I, Wiley Interscience, 1963.

7 Examples

If you have an assignment or report (or even thesis) to write, you may want to work on this and get help from our friendly volunteers. Otherwise, find a (very) short mathematical paper, such as

- *A One-Sentence Proof That Every Prime* $p \equiv 1 \mod 4$ *is a Sum of Two Squares*.
- *A Simple Proof that* π *is Irrational*.

and then see if you can reproduce it exactly, including the formatting and bibliography!