Example LaTeX Document
Demonstrating A Selection Of Features Most Arbitrary

Oliver Thomson Brown

Contents

1 General Text

There are a few things relating to simply writing text in LaTeX that probably bear mentioning. First of all quotes – simply using the single quote mark key on your keyboard leads to 'this'. To get an opening quote mark you actually need to use the backtick key which is just left of the '1' key on a UK keyboard. Additionally, one should never use the double quote key, as "this" happens. The proper procedure is two backticks to open, and two single quote marks to close – "wonderful". What if you need to use both? Then you need to use `\,` to separate them which typesets a thin space. "It must be admitted that certain things in LaTeX are quite 'tiresome'".

Moving on to dashes – did you know there are actually three?! The one I've been using so far, the humble em dash '–' is typeset by `--` and is a long pause, useful for separating sentences and clauses. Then there's the hyphen, used to combine words like 'Heriot-Watt', which is typeset by `-`. Finally there is the en dash '—', which apparently means 'through' as in 'pages 11—15', and I don't think I have ever bothered to use. So there you go.

The final thing I can think to mention here is paragraphs. Compiling the document you will see that each of these paragraphs after the first is tab indented, though inspection of the source will show you that all the text is left-aligned with a blank line between. In general LaTeX ignores whitespace, so if I introduce a line-break and tab indent in the source code here: LaTeX just ignores it.

On the other hand if I leave a blank line between the previous sentence and this one I am asking LaTeX to treat this as a separate paragraph, and essentially requesting that it do 'the right thing'. One can request a line-break using a double backslash.
Like so.

2 Mathematics

As mentioned in the presentation there are two main 'math(s) modes', inline and display. Inline is accessed with the delimiters `\(` and `\)`, and produces unlabelled inline equations like $M = USV^\dagger$. On the other hand display mode is invoked by beginning an equation environment with `\begin{equation}`, and produces separate labelled equations like,

$$\hat{H} = \sum_i^N \left[\Delta_{01} \hat{a}_i^\dagger \hat{a}_i + \left(\frac{\Delta_{02}}{2} - \Delta_{01} \right) \hat{a}_i^\dagger \hat{a}_i^\dagger \hat{a}_i \hat{a}_i + \frac{\Omega_D}{\sqrt{2}} \hat{a}_i \hat{a}_i + \frac{\Omega_D^*}{\sqrt{2}} \hat{a}_i^\dagger \hat{a}_i^\dagger \right] - J \sum_{\langle ij \rangle}^N \left[\hat{a}_i^\dagger \hat{a}_j + \hat{a}_i \hat{a}_j^\dagger \right], \tag{1}$$

where I have made liberal use of both the subscript operator `_{}`, and the superscript operator `^{}`. The amsmath package as well as various extra maths symbols also adds a couple of particularly useful features, the first of which is the align environment. It is another display math environment, but it allows one to easily split equations across multiple levels – useful for showing working out, and for particularly long equations like,

$$\begin{aligned}\dot{\rho}(t) = -i\left[\hat{H},\rho\right] + \sum_i^N \Big[&\frac{\gamma_{01}}{2}\left(2\hat{a}_i\hat{a}_i\hat{a}_i^\dagger\rho\hat{a}_i\hat{a}_i^\dagger\hat{a}_i^\dagger - \hat{a}_i\hat{a}_i^\dagger\hat{a}_i^\dagger\hat{a}_i\hat{a}_i\hat{a}_i^\dagger\rho - \rho\hat{a}_i\hat{a}_i^\dagger\hat{a}_i^\dagger\hat{a}_i\hat{a}_i\hat{a}_i^\dagger\right) \\ &+ \frac{\gamma_{12}}{2}\left(2\hat{a}_i^\dagger\hat{a}_i\hat{a}_i\rho\hat{a}_i^\dagger\hat{a}_i^\dagger\hat{a}_i - \hat{a}_i^\dagger\hat{a}_i^\dagger\hat{a}_i\hat{a}_i^\dagger\hat{a}_i\hat{a}_i\rho - \rho\hat{a}_i^\dagger\hat{a}_i^\dagger\hat{a}_i\hat{a}_i^\dagger\hat{a}_i\hat{a}_i\right)\Big], \end{aligned} \tag{2}$$

where `&` has been used to mark where the equations should be made to align (at the $+$), and the usual `\\` marks the linebreak. The command `\notag` has been used to suppress the label from the first line, but it is possible to have every line labelled independently, for example if one wants to express a series of simultaneous equations.

Another highly useful feature added by amsmath is the matrix environment, which allows one to easily typeset a matrix with a particular set of delimiters. The usual parentheses delimited matrix for example, is given by `\begin{pmatrix}`. It looks like this,

$$\hat{H} = \begin{pmatrix} 0 & 0 & \Omega_D \\ 0 & \Delta_{01} & 0 \\ \Omega_D^* & 0 & \Delta_{02} \end{pmatrix}, \tag{3}$$

which is the single-site Hamiltonian from eq. (1).

3 Graphics

For graphics the graphicx package must be used. Create a `figure` environment and use `\includegraphics[width=\l` to include the image. A caption can be added with `\caption{}`, and as ever `\label{}` can be used to provide a handle for cross-referencing. Options can be provided to the figure environment to *request* more specific placement of the figure. The one you'll most likely want is `[h!]` which says "ignore everything else, please place the figure *here*". LaTeX may or may not respect this request...

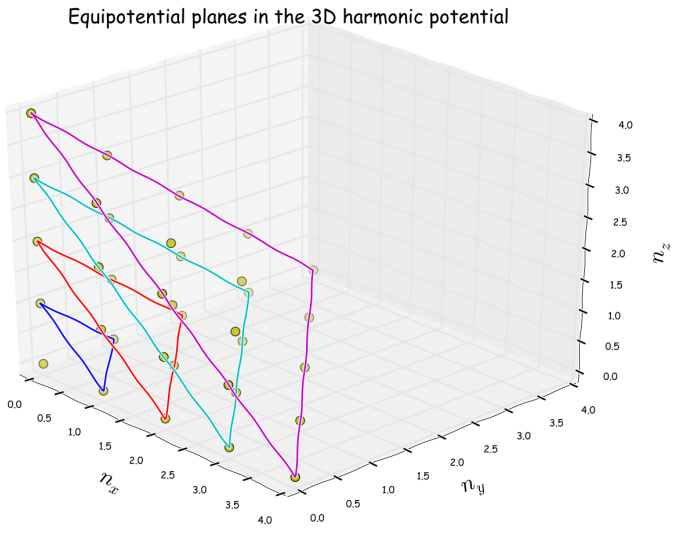

Figure 1: Did you know there's a context manager for matplotlib that lets you create plots that look like the webcomic xkcd? Well now you do! [1, 2]

4 References

First of all cross referencing! If one wants to refer to the image in this document there are two options – `\ref{label}`, and `\cref{label}`. The first is a built in LaTeX command, it works fine but is a bit clunky as it requires you to provide your own reference descriptor. To reference the image above, for example, one writes `fig.~\ref{fig:1}` which typesets fig. 1. On the other hand one can use the cleveref package, which automatically identifies the environment you're referencing and describes it appropriately. The commands `\cref{fig:1}`, `\cref{eq:1}`, and `\cref{sec:1}`, typeset fig. 1, eq. (1), and section 1 respectively. These labels are also fully customisable. I recommend the cleveref package.

To cite references using BibTeX one needs to include `\bibliographystyle{}` somewhere in your source file, the command `\bibliography{}` where you want the reference list to be placed, and to use the `\cite{}` command to place citations. The argument supplied to the `\cite{}` command is the citation label or key, and is the text found on the first line of the bib entry. For example in *example_ OTB.bib* you will find the entry for *Numberical Recipes* begins `@book{NR,` so 'NR' is the citation key, and the command `\cite{NR}` produces the citation [3].

Bibliography styles are given by .bst files and dictate both the citation style and the reference list style. Here I am using a modified version of the apsrev4-1 style provided by the American Physical Society for their publications (they actually even have their own 'revtex' document class!). The .bib file itself is simple enough to write by hand for only a few references (or for references to websites), but many journals will let you download a bibentry directly. Referencing software such as Mendeley and EndNote can also be used to automatically create .bib files. For EndNote one needs to install a plugin which can be found at `http://endnote.com/downloads/style/bibtex-export`, and be sure to write a citation key into the 'Label' field on the EndNote entry! Mendeley will generate a citation key for you in the format [First-author-surname][year]. For example `\cite{Brown2013}` references a paper exported from Mendeley in example_OTB.bib [4].

The only other thing you need to know about referencing is that the document needs to be compiled twice for cross-referencing (pdflatex-pdflatex), and four times for referencing with BibTeX (pdflatex-pdflatex-bibtex-pdflatex)! Most IDEs will let you do this with a single button press.

References

[1] R. Munroe, "xkcd," `http://xkcd.com/` (2016), [Online; accessed 2016-01-20].

[2] Matplotlib Development Team, "matplotlib," `http://matplotlib.org/` (2016), [Online; accessed 2016-01-20].

[3] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes – The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, 2007).

[4] O. T. Brown, J. Truesdale, S. Louchart, S. McEndoo, S. Maniscalco, J. Robertson, T. Lim, and S. Kilbride, "Serious game for quantum research," Serious Games Development and Applications , 178 (2013).