How to install and use LaTeX

Dionne M. Aleman* Dionne M. Aleman†

September 17, 2022

Abstract

This is a quick and dirty guide to doing most of the things most people will ever want to do with LaTeX. This guide contains a number of examples of useful things (e.g., abstracts, arrows, tables, etc.) that are not explicitly described. Just look at the source `.tex` file to see how it was done (that is why there are a number of random bullets, symbols, sections, formatting styles, etc. throughout the guide). This document is certainly not the definitive LaTeX manual; if there is something you want to do and it is not described here, look it up online. There are no limits to what can be done!

*Department of Mechanical and Industrial Engineering, University of Toronto; 5 King's College Road, Toronto, ON M5S 3G8, Canada; `aleman@mie.utoronto.ca`

†Department of Health Policy, Management and Evaluation, University of Toronto

Contents

1 What is LaTeX?

From Wikipedia (2008), TeX (pronounced "tek") is a typesetting system created by Donald Knuth. Together with the METAFONT language for font description and the Computer Modern typeface, it was designed with two main goals in mind: to allow anybody to produce high-quality books using a reasonable amount of effort, and to provide a system that would give the exact same results on all computers, now and in the future. Within the typesetting system, its name is formatted as TeX.

The key to being a proficient LaTeXuser is simple: Stop worrying about formatting, and just trust that LaTeX will do it right for you.

1.1 Why use LaTeX?

LaTeX can be difficult to learn, and most people are too hopelessly attached to Microsoft's products to try something new. If Word works pretty well, why change? Here's a breakdown of some of the more obvious pros and cons of LaTeX.

1.2 Pros and cons

Pros LaTeX documents are beautiful. The typesetting, spacing and kerning are all perfect, and the font (Computer Modern) is infinitely scalable, so it looks sharp at any size. As shown in this document, you can create links to sections, tables, figures, citations, equations, bullets, URLs; moreover, it's automatic! You will never again have to redo your section numbers because you reorganized a paper, or worse, redo your citations because you added in a new reference that throws off the numbering. You also won't have to redo your citations because the journal you are submitting your paper to wants a different citation format than what you used.

For a strict comparison to Word, TeX documents are just text, so they will never become bloated in file size (no matter how large your paper) or mysteriously crash and cause you to lose hours of work. LaTeX won't make your pictures suddenly disappear while you tried to move the image one pixel upward. Your equations will look elegant and respectable, and not like they were written by a child. Further, equations are *easy*—no more hunting for symbols and brackets in MathType. Tables are pure text, so you can automate table creation right from your data matrices in Matlab, C++, etc.

Cons LaTeX has a learning curve. Unlike Word, you can't really just start using it without spending some effort to learn what is going on. That's why we have this tutorial. Also unlike Word, LaTeX won't let you (easily) do whatever you want in your document. **The driving vision behind LaTeX is that *it* will do your typesetting; you just do the typing.** You are virtually not allowed to spend your time placing figures and tables exactly where you want them, because LaTeX will generally put things where it thinks they go best. At first, this causes a lot of users headaches because they want to nitpick over positioning (and usually in an incorrect fashion). Eventually, you will realize that LaTeX is in fact putting your floating objects in the best place they should be, and you will be happy not to have to worry about such things anymore.

Possibly the biggest drawback to LaTeX is that it does not have the reviewing toolbar that Word has, which allows you and collaborators to track changes made to a document. There are some ways to get around this if you are working with non-LaTeX users (one way it to copy the paragraph you want to change into Word and then track the changes made), but nothing is very elegant.

2 How to install LaTeX and editors

In Section 1, we learned about LaTeX. In Section 2.2, we'll talk about how to get and install LaTeX.

There are many different packages and editors available. The ones listed below are just ones I have used and liked, but there are certainly many other good ones. Most packages and editors are 100% free. Just download and install the packages and editors, and you are ready to go.

PRO TIP: Install LaTeX and any additional items first, then install your editor last. That way, your editor will probably auto-detect everything on its own.

2.1 Mac OS X 10.3 and higher

1. LaTeX package
   - MacTex: http://www.tug.org/mactex/
2. Editor
   - TeXShop: http://www.uoregon.edu/~koch/texshop/

2.2 Windows OS

- LaTeX package
  - MikTex: http://miktex.org/
- Additional items needed to handle images and `dvi` previews
  - Ghostscript: http://pages.cs.wisc.edu/~ghost/
  - GSview: http://pages.cs.wisc.edu/~ghost/
- Editor
  - TeXnicCenter: http://www.texniccenter.org/
  - WinEdt: http://www.winedt.com/
    - Personal opinion: Clumsy, but might be better with new v6.0.

2.3 How to use the editors

Once you have written your LaTeX document[1], you will "typeset" (compile) your work. Your editor will automatically create a dvi, ps or pdf document, depending on the settings you choose. On a Windows system, it is faster to create a dvi file than a pdf, especially when images are involved, so it is general practice to compile your document as a dvi file until everything is finished, and only then compile it into a pdf. On a Mac, TeXShop will compile and display a pdf as fast as a dvi, so there is no need to bother with dvi files.

3 How to write your LaTeX document

The rest of this document describes the basics of writing a paper in LaTeX. This covers everything that 90% of users will ever need, but only scratches the surface of what you can do. A famous comprehensive guide, "The Not So Short Introduction to LaTeX", details just about everything and can be found at

http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf

3.1 How to write mathematical equations

LaTeX makes writing mathematical equations ridiculously easy, and the results are picture-perfect.

3.1.1 Single equations

You can write a single equation one of two ways:

1. `\begin{equation} ... \end{equation}`
2. `\[ ... \]`

where "..." represents the equation itself. Method 2 is essentially just shorthand for Method 1 (look at the source document to see how these linked labels were made). For example, using Method 1:

$$F(\Theta) = \mu + \epsilon(\theta) \tag{1}$$

yields the same result as Method 2:

$$F(\Theta) = \mu + \epsilon(\theta)$$

Note that Method 2 does not number the equation. Equation 1 is numbered.

Here is an example of a fraction:

$$\frac{numerator}{denominator}$$

But, if you look closely, you can see that spacing between the letters in "numerator" and "denominator" is a bit off. The reason for the slightly inconsistent spacing is that LaTeX

[1] Look! A footnote! Did you notice that the footnote number in the text was a link to this footnote?

assumes that each letter is a variable, rather than the whole numerator/denominator being an actual word. We can fix this issue by specifying that regular text words should be written as regular text with the `\text{}` environment:

$$\frac{\text{numerator}}{\text{denominator}}$$

Pro tip When writing equations, **do not skip lines** in your source code. A skipped line in the source document is interpreted as a new paragraph, and if you skip a line before your equation, there will be too much vertical spacing before the equation in the `pdf`. Skipping lines after the equation is fine if you intend whatever content comes next to be a new paragraph.

3.1.2 Multiple equations

There are two ways to write multiple equations: `eqnarray` and `align`. The major difference between the two is the method of alignment and the numbering options. I personally have a strong preference for `align` as it is visually easier to understand.

eqnarray The first method is to use the `eqnarray` environment:

```
\begin{eqnarray} ... \end{eqnarray}
```

This environment basically sets up three columns: the first is right-justified, the second is centered and the third is left-justified. This allows for clear presentation of mathematical formulas.

The columns are separated by "`&`", and a "`\\`" must be placed at the end of each equation to signify the end of the line. To align the equations, say, along the equals sign, place ampersands (`&`) around the equals sign in each equation. Here is an example mathematical model using `eqnarray`:

$$\begin{aligned}
\text{minimize} \quad & -\underline{\sigma\tau}_n^{2m}\left\{1-\mathbf{r}^\top\mathbf{R}_n^{-1}\mathbf{r}+\frac{\left(1-\mathbf{e}^\top\mathbf{R}_n^{-1}\mathbf{r}\right)^2}{\mathbf{e}^\top\mathbf{R}_n^{-1}\mathbf{e}}\right\}+\textstyle\sum_{i=1}^n\left(w_i^L\right)^2 & \\
\text{subject to} \quad & (a_i^c+b_i^c r_i)+c\textstyle\sum_{h=1}^k\left(p_{i,h}^t+q_{i,h}^t\theta_h\right)\le 0 & \forall i=1,\ldots,n \\
& (a_i^t+b_i^t r_i)+c\textstyle\sum_{h=1}^k\left(p_{i,h}^c+q_{i,h}^c\theta_h\right)\le 0 & \forall i=1,\ldots,n \\
& w_i^L\neq 0 & \forall i=1,\ldots,n \\
& w_i^L+r_i^L\le r_i & \forall i=1,\ldots,n \\
& w_i^U\le 0 & \forall i=1,\ldots,n \\
& w_i^U-r_i\le r_i^U & \forall i=1,\ldots,n \\
& \ell_h\le\theta_h\le u_h & \forall h\in\mathcal{S}, h\notin\mathcal{T}
\end{aligned}$$

align The second method, the `align` environment, is similar, and the major difference is the column alignment. While `eqnarray` using right-center-left alignment, the `align` environment using right-left-right-left-..., for however many columns you chose to use. Also, there is no default spacing between the columns. The `align` environment makes it easy to add tags to

your equations rather than numbers. For example, say the above optimization problem is named BOO-EI We can tag it using the `align` environment:

$$\text{minimize} \quad -\underline{\sigma\tau}_n^{2m} \left\{ 1 - \mathbf{r}^\top \mathbf{R}_n^{-1} \mathbf{r} + \frac{\left(1 - \mathbf{e}^\top \mathbf{R}_n^{-1} \mathbf{r}\right)^2}{\mathbf{e}^\top \mathbf{R}_n^{-1} \mathbf{e}} \right\} + \sum_{i=1}^{n} \left(w_i^L\right)^2 \tag{BOO-EI}$$

$$\begin{aligned}
\text{subject to} \quad & (a_i^c + b_i^c r_i) + c \sum_{h=1}^{k} \left(p_{i,h}^t + q_{i,h}^t \theta_h\right) \leq 0 && \forall i = 1, \ldots, n \\
& \left(a_i^t + b_i^t r_i\right) + c \sum_{h=1}^{k} \left(p_{i,h}^c + q_{i,h}^c \theta_h\right) \leq 0 && \forall i = 1, \ldots, n \qquad (2) \\
& w_i^L \neq 0 && \forall i = 1, \ldots, n \qquad (3) \\
& w_i^L + r_i^L \leq r_i && \forall i = 1, \ldots, n \qquad (4) \\
& w_i^U \leq 0 && \forall i = 1, \ldots, n \qquad (5) \\
& w_i^U - r_i \leq r_i^U && \forall i = 1, \ldots, n \qquad (6) \\
& \ell_h \leq \theta_h \leq u_h && \forall h \in \mathcal{S}, h \notin \mathcal{T} \qquad (7)
\end{aligned}$$

Now we can refer to whole problem BOO-EI (rather than individual equations) using a linked reference (`\ref{eqn:boo-ei}`)! Note that we can manually specify which equations are labeled by putting "`notag`" at the end of any line that we do not want numbered/tagged, like the first constraint.

3.1.3 Numbering equations

Equations are automatically numbered, but if you don't want to number an equation, either use the shorthand method or place an asterisk (`*`) after `eqnarray` (i.e., `eqnarray*`) and `align` (i.e., `align*`) in both the begin and end statements. This also works for suppressing numbers in sections. For example:

$$\begin{aligned}
\hat{\mathbf{x}} &= \bar{w} + \underline{y} + z \\
\text{mass} &= \text{density} \times \text{volume}
\end{aligned}$$

$$\hat{\mathbf{x}} = \bar{w} + \underline{y} + z \tag{8}$$

$$\text{mass} = \text{density} \times \text{volume} \tag{9}$$

3.1.4 Matrices/arrays

Matrices are just arrays displayed with brackets around them. For example:

$$\mathbf{R} = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}, \tag{10}$$

$$\mathbf{R} = \begin{pmatrix} a_1 & \cdots & z_1 \\ \vdots & \ddots & \vdots \\ a_n & \cdots & z_n \end{pmatrix}. \tag{11}$$

Dots are created using `\cdots` (center dots), `\vdots` (vertical dots) and `\ddots` (diagonal dots). Ellipses-style dots are created by `\ldots` (lower dots).

Using `\left(` and `\right)` will create parentheses that expand to the appropriate size to fit the content inside them. The (and) can be replaced with any other bracketing-type symbol with the same effect. A `\left` must ALWAYS be accompanied by a `\right`. To only use the left or right bracket/brace/etc., use a "." instead of the omitted bracket/brace/etc. symbol.

$$y_i = \begin{cases} 1 & \text{if facility is built at location } i \\ 0 & \text{otherwise} \end{cases}$$

```
\begin{equation*}
y_i = \left\{ \begin{array}{ll}
1 & \text{if facility is built at location } i \\
0 & \text{otherwise}
\end{array} \right.
\end{equation*}
```

3.1.5 Greek letters

If you can spell the names of Greek letters, you can easily use them in LaTeX.

Lowercase letters Lowercase Greek letters are simply \name, where "name" is the name of the letter in all lowercase. For example:

$$\begin{aligned} \backslash\text{gamma} &\rightarrow \gamma \\ \backslash\text{delta} &\rightarrow \delta \\ \backslash\text{pi} &\leftarrow \pi \\ \backslash\text{phi} &\leftarrow \phi \end{aligned}$$

Uppercase letters Uppercase Greek letters are simply \Name, where "Name" is the name of the letter in with the first letter capitalized. For example:

$$\begin{aligned} \backslash\text{Gamma} &\Rightarrow \Gamma \\ \backslash\text{Delta} &\Rightarrow \Delta \\ \backslash\text{Pi} &\Leftarrow \Pi \\ \backslash\text{Phi} &\Leftarrow \Phi \end{aligned}$$

Table 1: Minimum objective function values and time computation time.

	Obj. fn. value		*Run time (hrs)*	
case	Avg.	St. Dev.	Avg.	St. Dev.
1	565.24	8.82	5.35	5.07
2	570.51	12.83	7.49	3.78
3	893.45	20.60	7.05	2.21
4	710.92	7.72	6.54	3.39
5	512.22	20.04	6.96	3.33
6	799.95	34.07	3.48	3.40

Table 2: Minimum objective function values and time computation time.

	Obj. fn. value		*Run time (hrs)*	
case	Average	St. Dev.	Average	St. Dev.
1	565.24	8.82	5.35	5.07
2	570.51	12.83	7.49	3.78
3	893.45	20.60	7.05	2.21
4	710.92	7.72	6.54	3.39
5	512.22	20.04	6.96	3.33
6	799.95	34.07	3.48	3.40

3.1.6 Math inline with text

To use math inline with text, type the equation inline with your text, and surround it by \$...\$. The equation will automatically be formatted to best fit in the paragraph. For example, with $\sum_{i \in S} x_i \geq 1$, the subscript for the summation is placed to the side rather than underneath.

```
For example, with $\sum_{i \in S} x_i \geq 1$, the subscript for the summation
is placed to the side rather than underneath.
```

3.2 Tables

Tables are a little bit of work in LaTeX. Table 2 uses a plain layout. Alternatively, you can use the table styles defined in the `booktabs` package for a more polished look as in Table **??**.

Although the creation of tables is not as intuitive as in, say, Microsoft Excel, the pure text format certainly has its advantages. For example, if you have a matrix of values in Matlab (or any other programming environment/language) that you want to put in a table in your document, you don't have to copy those values over by hand. Just write a quick function that will read in the matrix and print out the ampersands (`&`) and endlines (`\\`) appropriately. Regardless of how large your table is, you just have to call one function and your table is ready for you to paste into your `.tex` file!

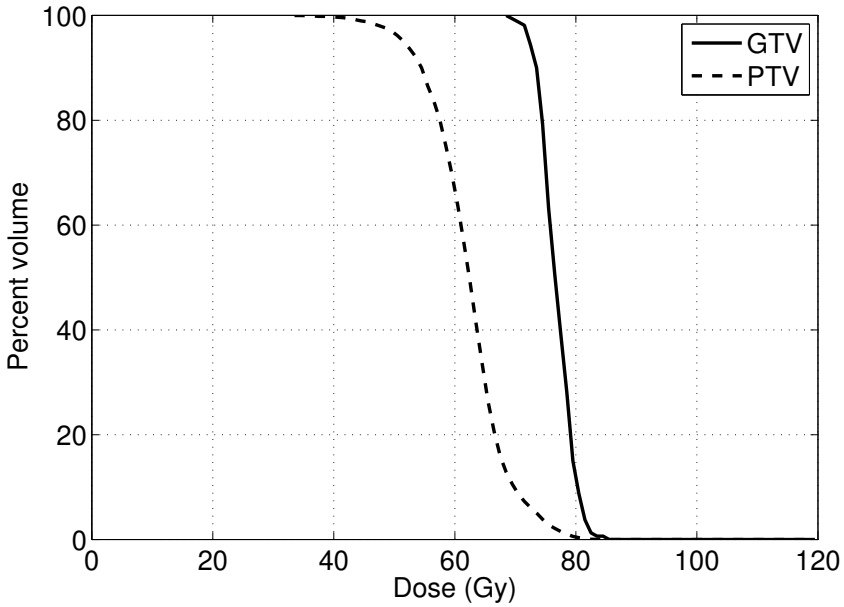


Figure 1: Graphical results

3.3 Figures

Figures are fairly easy, and new LaTeX distributions no longer mandate eps or pdf format (though eps and pdf are generally better for scaling). You can save your figures in Photoshop as eps files, but make sure to deselect all options. Figure 1 may not be pretty in black and white, but most journals don't print in color. So, make sure your graphics are legible in black and white. Also make sure that any font in your figure (e.g., legends) is legible. This usually means that the fonts will have to be seemingly unreasonably large.

We can also easily create subfigures using the `subfigure` package. We can then refer to them as Figure 2(a) and Figure 2(b), or refer to the whole figure set as Figure 2.

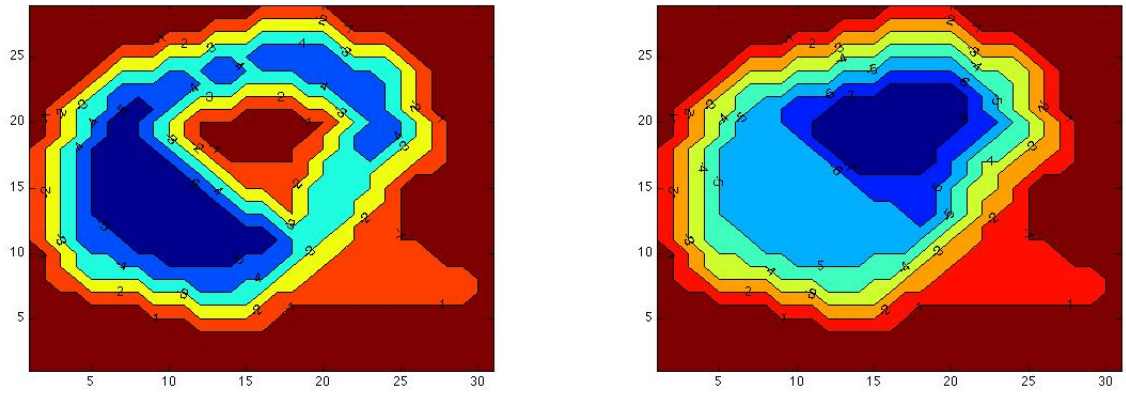
Look at the code to generate Figures 1 and 2 in the `tex` file. Note that the width can be specified as a percentage of the total text width (`[width=0.75\textwidth]`) or as a scale of the original figure size (`[scale=0.4]`).

3.4 Citations

Citations in LaTeX are done through another package called BibTeX, which should come pre-installed with your LaTeX package. Simply write the information for all the publications you want to cite in a text file with extension `.bib`, and then include that file (without extension) and a particular citation style as a bibliography at the end of your document:

```
\bibliographystyle{plain}
\bibliography{bibfile_latex_howto}
```

See the `bibfile_latex_howto.bib` document for a template on how to format the `.bib` file. The first line of every reference in the `.bib` file is the key/tag used to refer to that



(a) Contour plot of iteration k

(b) Contour plot of iteration $k + 1$

Figure 2: Contour plots of the target at two subsequent iterations of the skeletonization algorithm.

citation. To reference a single publication (Aleman et al., 2009), simply write the citation's key inside a `\cite{}` command:

```
\cite{aleman-joc-2009}
```

For multiple citations like Aleman et al. (2008, 2009); Jones et al. (1998):

```
\cite{aleman-jogo-2008,aleman-joc-2009,jones-1998}
```

Unless told otherwise, BibTeX will order your bibliography by last name of first author and then by year. Section 3.4.1 shows how to automatically sort and compress your citations.

What if you want your citations to be something other than square brackets with the citation number? Section 3.4.2 describes how to customize your citation styles.

3.4.1 Sorting and compressing citations

It is generally considered very bad form to have your citations appear in non-alphabetical or non-numerical order. But, it can be tedious to re-arrange your citations or insert new citations if you have a string of several reference. Compressing citations (e.g., [1–4,7] instead of [1,2,3,4,7]) can also be time-consuming. Fortunately, the `natbib` package will automatically take care of that.

Just include the `natbib` package in your header with options to either sort or sort and compress:

```
\usepackage[sort]{natbib}
\usepackage[sort&compress]{natbib}
```

Note that compressing will only change the citation appearance if you are using numbered citations.

Table 3: Selected citation options. Note that not all citation options will work with the `plain` bibliography style.

Citation code	Output
`\cite{aleman-joc-2009}`	Aleman et al. (2009)
`\citep{aleman-joc-2009}`	(Aleman et al., 2009)
`\citep*{aleman-joc-2009}`	(Aleman, Romeijn, and Dempsey, 2009)
`\citet{aleman-joc-2009}`	Aleman et al. (2009)
`\citet*{aleman-joc-2009}`	Aleman, Romeijn, and Dempsey (2009)
`\citeauthor{aleman-joc-2009}`	Aleman et al.
`\citeauthor*{aleman-joc-2009}`	Aleman, Romeijn, and Dempsey
`\citeyear{aleman-joc-2009}`	2009

3.4.2 Fancier citations

For fancier citations (e.g., "name, year" format), use the `natbib` package and change the bibliography style from `plain` to something else. Rather than make your own style, it easier to find a journal whose style you like, and then download their citation style (which is usually publicly available). The style file used here is from the INFORMS Journal on Computing, and is contained in file `ijocv081.bst`.

You can create your own bibliography style from scratch by typing `latex makebst` in a terminal window. Just follow the prompts to define your style.

When you use the "name, year" format, you will often want to place your citation in parentheses. For example, "The first use of an unrestricted optimal beam selection algorithm employed the response surface method (Aleman et al., 2009)". If you want your citations to appear in parenthesis, use `\citep` (Aleman et al., 2009). If you want to use an in-text citation, try `\citet`. For example, Aleman et al. (2009) showed that their research is better than everyone else's.

Although `\cite` was initially used in this tutorial for pedagogical purposes, I recommend you use `\citep` or `\citet` instead. It does not make a difference when using the `plain` bibliography style (since all citations are formatted like [1]), but it will make a difference if you use a fancier citation style. Table 3 demonstrates a number of possible citation options. Note that not all citation options will work with the `plain` bibliography style.

3.4.3 Compiling citations

To get your references to show up, you will have to:

1. Typeset your document with LaTeX
2. Typeset your document with BibTeX
3. Typeset your document with LaTeX twice more.

The first step tells the editor which publications from your `.bib` file you want to cite; the second step formats the citations according to your bibliography style; and then the third

step figures out what numbering each citation should have (usually based on last name of first author) and puts it all in your document.

TeXnicCenter actually compiles both LaTeX and BibTeX simultaneously, so you effectively have to compile the document 3-4 times to get the in-text citations updated.

The same principle of multiple compilations applies to generating the table of contents and figure/table numbers, as well as to section, equation, table and figure referencing.

3.4.4 Making your citations into links

Your references to bibliography items and to other sections can be turned into actually hyperlinks in the document by using the `hyperref` package. Nothing different needs to be done in terms of citing references or referring to sections, tables or figures; just include the package in the document header.

See the source code of this document for an example of how to set colors.

4 LaTeX instead of PowerPoint

LaTeX can be used a presentation medium as well through a package called Beamer:

http://bitbucket.org/rivanvx/beamer/wiki/Home

One of the great benefits of Beamer, aside from preventing the use of animations, is the dynamic presentation outline that allows your audience to see exactly where in the presentation you are at any given moment.

More information on how to use Beamer can be found at the link above, but it basically boils down to putting

```
\begin{frame}
\frametitle{Your frame title}
...
\end{frame}
```

around your content.

References

Aleman, D.M., A. Kumar, R.K. Ahuja, H.E. Romeijn, J.F. Dempsey. 2008. Neighborhood search approaches to beam orientation optimization in intensity modulated radiation therapy treatment planning. *Journal of Global Optimization* **42** 587–607.

Aleman, D.M., H.E. Romeijn, J.F. Dempsey. 2009. A response surface approach to beam orientation optimization in IMRT treatment planning. *INFORMS Journal on Computing: Computational Biology and Medical Applications* **21** 62–76.

Jones, D.R., M. Schonlau, W.J. Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* **13** 455–492.

Wikipedia. 2008. http://en.wikipedia.org/wiki/TeX.

A Tables of data

Look, an appendix! You can reference an appendix just like anything else. See the code in Appendix A for examples.

B Figures upon figures

Another appendix! This one is automatically named Appendix B.