A small example of report

Olivier Buffet

July 31, 2018

Abstract

This document is meant to be short and self-explanatory. Please (i) look at the LaTeX file for details of how to do things, and (ii) be curious and search for documentation (e.g., starting with the references at the end of this document).

Chapter 1

Chapter title

1.1 Section title

This document is meant to be short and self-explanatory. Please (i) look at the LaTeX file for details of how to do things, and (ii) be curious and search for documentation.

1.1.1 Compiling

First things first: a LaTeX document is somehow like a program. You need to "compile" it to obtain a pdf document. Here, you can simply get a pdf file by typing "`make`" in a shell (assuming you are under Unix).

Type "`make clean`" to clean up everything. This may be necessary if a corrupted `*.aux` file has been generated.

1.1.2 Editing

You can edit a LaTeX source file with a simple text editor (e.g. vim or emacs; no word processor please), but may want to use a LaTeX IDE.

Headings

This is just to show you that you have access to various levels of headings. In an article you will often use sections, subsections and subsubsections (but could also use paragraphs and subparagraphs).

1.2 More Advanced Stuff

1.2.1 Maths

Some text just to give an example with words in *italic* or in **bold** fonts, and some formulas:

$$Q(s,a) = \sum_{s' \in S} T(s,a,s')[c(s,a,s') + V(s')]. \tag{1.1}$$

Please, note that a math formula is part of the text (just written in a specific way). Thus, use punctuation in formulas adequately (see the dot at the end of Formula 1.1).

1.2.2 Inserting Floating Objects

Figures, algorithms and tables are floating objects, i.e., you should never attempt to place them at a precise point in the body of your text. LaTeX will take care of placing everything nicely. Thus, you should always make at least one reference to each floating object in the body of the text. If needed, you can still give some recommandations through options like "`[htbp]`", here meaning: *"try to place the float* `h`*ere first, otherwise at the* `t`*op of the page, otherwise at the* `b`*ottom, or else on a separate* `p`*age"*.

We now give examples of how to make a simple figure, table and algorithm. There should always be at least one reference to each floating object in the body of your document. In this case, the name of this object becomes a proper name and deserves a capital letter as in the following examples: Figure 1.1, Algorithm 1 and Table 1.1.

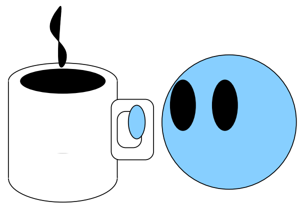

Figure 1.1: Example figure

1.2.3 Labels, References and Citations

Note that you can refer to a variety of objects within your document like: floating objects, equations/formulas, numbered lines in an algorithm, chapters/sections/subsections ... Just put a label where needed. Note that you have to refer to Section 1.1.2 the first time, but can use Sec. 1.1.2 afterwards.

You will probably also want to cite bibliographical references like this: [KLM96, Rab89]. To that end, you need a bib file, i.e., a small database with an entry for each document you want to refer to. Please learn to fill these entries properly.

To go further with bibliographies:

- Note that you can choose the bibliographic style to be used: alpha, num, unsrt...
- A bibliographic reference of the form "[XX]" or "(XX)" is a parenthesis. One should thus never write "as proved by [XX]" or "please refer to (XX)". Using packages like natbib (and its command `\citet{...}`) allows writing, e.g., "as proved by Bellman (1953)", which is appropriate. A good starting point on this topic is: `http://merkel.zoneo.net/Latex/natbib.php`.

Algorithm 1: Goal Reachability

GoalReachability(G_R: reachability graph) ;
all states initially marked as `trapped` ;
for $s \in G_R$ *s.t.* *GOAL*(s) **do**
    FindNotTrapped(s);
for $s \in G_R$ *s.t.* *trapped*(s) **do**
    for $s' \in Parents(s)$ *s.t.* $\neg$(*trapped*(s') $\vee$ *visited*(s')) **do**
        FindDangerous(s') ;
FindNotTrapped(s: state) ;
UnMark(s, `trapped`) ;
for $s' \in Parents(s)$ *s.t.* *trapped*(s') **do**
    FindNotTrapped(s') ;
FindDangerous(s: state) ;
Mark(s, `visited`) ;
if $\forall a \in A,\ \exists s' \in Sons(s)$ *s.t.* $Pr^{\max}(s'|s,a) > 0$ **then**
    Mark(s, `dangerous`) ;
    for $s' \in Parents(s)$ *s.t.* $\neg$*visited*(s') **do**
        FindDangerous(s', s) ;

	a	b	c
a	aa	ab	ac
b	ba	bb	bc
c	ca	cb	cc

Table 1.1: Table example

1.2.4 Drawing Figures

Which Format? Which Tool?

Given a figure you want to draw and include as a figure in a LaTeX document, you have to choose (i) a software tool and (ii) a file format. Here is a classification of the main cases you may encounter:

vector: if your figure is essentially geometric, the best is to use a *vector* drawing tool, each having its own working file format, such as:

- **xfig,** which remains, IMHO, a very good tool because of its many possibilities (e.g. including math formulas); it suffers a bit from a somewhat old-fashionned interface;
- **inkscape,** which is very powerful but, IMHO, more intended for vector art;
- **pgf/tikz,** a LaTeX package allowing to describe a picture in a specific language.

For use with pdfLaTeX, you should usually export your image in pdf format (except for pgf/tikz).

bitmap: for non-geometric figures—e.g. numerized photographs or pictures, but also vector drawings with advanced effects (blur, transparency...)—it is more appropriate to use a *bitmap* format, that is, a format representing the image as a map of points; a vector drawing tool can produce images in bitmap format, but they can also be produced with softwares such as The Gimp; for pdfLaTeX, you should typically use:

- **png** for "drawing like" images, especially because it preserves areas with a uniform color;
- **jpg** for photographs, because of its more appropriate compression algorithm.

Maths Formulas / LaTeX code

The only tool I know for putting LaTeX code—e.g., math formulas—as text in a vector image is xfig. There are various ways to produce your image and include it in LaTeX, but here is a simple process:

1. in your xfig image, type your LaTeX code as normal text (e.g., under '$'s for maths formulas);
2. for each text that needs to be processed by LaTeX, edit it (`edit` button) and put it in `Special` mode;
3. (save and) export your image in "`Combined PDF/LaTeX`" format;
4. in your LaTeX document, make sure to include the following packages:

```
\usepackage{amsmath,amssymb}
\usepackage{color}
\usepackage{graphicx}
```

5. again in your LaTeX document, include your image as follows (the image being here named “ex” and displayed with a .5 factor):

```
\begin{figure}[htbp]
  \centering
  \scalebox{.5}{
    \input{ex.pdf_t}
  }
  \caption{My sub-title
    \label{fig:exWithMaths}}
\end{figure}
```

which leads to the result of Figure 1.2.

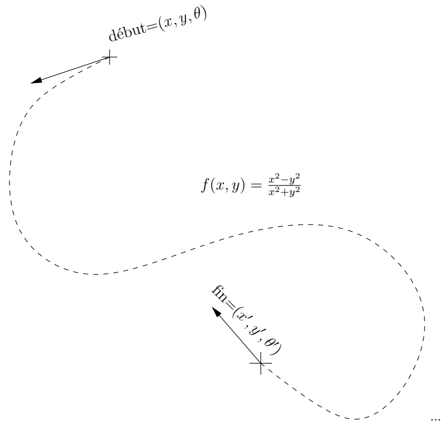

Figure 1.2: My sub-title

Note that you can specify the width and/or height of your picture with `\resizebox` instead of `\scalebox`, for example:

- `\resizebox{width}{height}{...}` to specify the width and height;
- `\resizebox{width}{!}{...}` to specify the width (the height being computed so as to preserve the aspect ratio).

Generating Plots and Curves, and Annonating Them

When not drawing things by hand or using numerized pictures, you may want to generate plots (from numerical data) and/or curves (from math formulas). Here are various tools for that:

- spreadsheet softwares such as MS-Excel, OpenOffice’s Calc or Gnumeric (some of them being also able to export LaTeX tables!), preferably for generating plots from numerical data,
- math toolboxes such as Matlab, Octave or Scilab, preferably for plotting curves from math formulas,
- Matplotlib from within python code,

- command-line tools like Gnuplot, preferably for everything.

Gnuplot has the advantage to be a very lightweight tool compared to its aforementionned colleagues. Plus, it can easily output image in xfig's file format, which makes it possible to add annotations (maths symbols, arrows pointing at particular points...).

Figures 1.3 and 1.4 give two examples of 2D curves (without annotations).

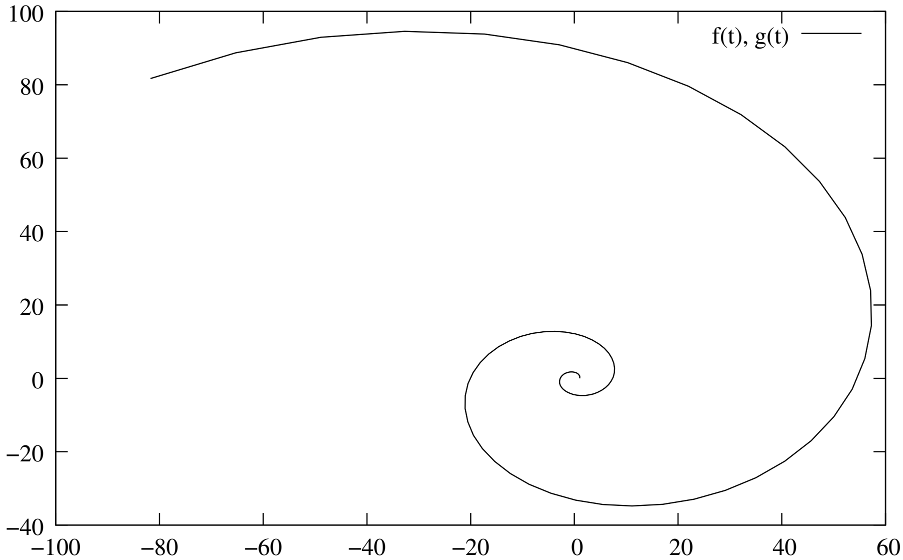

Figure 1.3: Logarithmic Spiral: $\rho(\theta) = e^{\theta}$

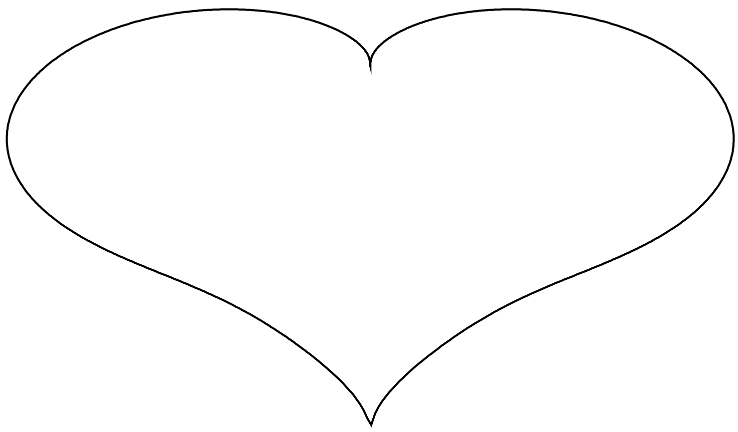

Figure 1.4: Cardiod: $\rho(\theta) = \frac{\sin(\theta)\sqrt{|\cos(\theta)|}}{\sin(\theta)+1.4} - 2\sin(\theta) + 2$, for $\theta \in [-\pi, +\pi]$

Bibliography

[KLM96] L. Kaelbling, M. Littman, and A. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[Rab89] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

Appendix A

Now In Appendix Mode

Appendices typically come after the bibliography. Note the different numbering!