MATH549 Exercise Sheet 1

Deadline for submission: Monday 6th October

Introduction

LaTeX is a professional-quality typesetting system, which produces better output than word processing programs (e.g. Microsoft Word), especially for mathematical documents. It takes a bit of effort to learn, but once you're familiar with it I hope you'll find that it's fast and easy to use.

Each week you will work through an exercise sheet similar to this one: the end result will be one or more files, which you should email to me by the submission deadline.

Page numbers in the text and in the margin of this exercise sheet refer to your *Not So Short Introduction to LaTeX 2ε*. Where a page number is given in the margin, it normally means that the Not So Short Introduction says substantially more about the topic than will be in the sheet. (The references are to version 4.26 of the Not So Short Introduction: if you download more recent versions, the page numbers may change.)

I recommend that you keep the sample LaTeX document that I distributed, and the code which produced it, to hand. In many cases, looking at a "real life" document of this sort is more helpful than reading though step-by-step instructions.

The website for this module is `http://www.liv.ac.uk/~tobyhall/latex/`. There you will find this and future exercise sheets, a link to the Not So Short Introduction, and other useful documents and links. I suggest that you bookmark this page. **Before you do anything else, please go to this webpage and follow the instructions given there for getting started with LaTeX in room 302.**

Getting Started

We shall be using the Windows program TeXShell to write and process LaTeX documents. Start up TeXShell by double-clicking on the desktop icon. You're now ready to create your first LaTeX document.

Exercise 1a

Choose `New` from the `File` menu to start a new document, and type the following in the edit window:

```
\documentclass{article}
\begin{document}
Here begins my first document written in \LaTeX.
\end{document}
```

Click on the `Save` button, and save the file in a suitable folder on your `M:` drive (perhaps a `LaTeX` subfolder of the `documents` folder). Please choose a filename by putting a `1` after your surname: thus if I were doing this, I'd call my file `hall1.tex`. Naming your file like this makes my life easier, since I can keep the submitted files for the whole class in one folder on my computer.

There should now be a button labelled with this filename just above the edit window. Right-click on this button, and choose `Set as main file` from the drop-down menu. This tells TeXShell that it is this file, and not any others you may happen to have open, that you want to process and view. Now click on the `pdfLaTeX` button. A window with a black background pops up and, hopefully, disappears again quickly. If it doesn't disappear, it means that you've made a mistake in your document. **Whenever this happens, don't just close the black window or shuffle it away behind TeXShell: enter `q` at the `?` prompt to close it down properly.** Having done this, check that your file is exactly as shown above. In particular, LaTeX is *case-sensitive*: that means that typing `\begin{Document}` or `\Latex` is a possible cause of error. Once you've got it to work, click on the `View pdf` button and marvel at the result.

There's a good deal to explain about what you've just done: first, though, a note on how this exercise sheet is going to work. Each numbered part of the sheet asks you to make some addition or modification to your file: at the end, this file is what you will submit to me. After each step, you should click on the `pdfLaTeX` and `View pdf` buttons to see the result of your changes. (Note that you need to close your PDF file in Acrobat Reader before clicking the `pdfLaTeX` button again: the easiest way to do this without closing Acrobat Reader itself is to hit `CTRL-W`. You can also hit `F6` and `F8` instead of clicking the `pdfLaTeX` and `View pdf` buttons if you find that easier.) You may also want to experiment with various commands as you work through the sheet — you could do this in a second file, perhaps called `Test.tex`, which you should also keep open throughout. Remember to use `Set as main file` when you change the file you want to process.

If you want to check what your document should look like when you've finished, you can look at the 'solutions' I handed out. However, **work from this instruction sheet, not from the solutions. If you simply try to reproduce the solutions, you will go wrong.** Note also that you will often be asked to modify things you've done earlier in the sheet, or to add things to the middle of your document, so the solutions are not a good guide to what your document should be looking like as you work through the sheet.

What's happening when you press the `pdfLaTeX` and `View pdf` buttons? The `pdfLaTeX` button runs the LaTeX program, which converts the `tex` file you've typed into a PDF file. The `View pdf` button just loads this PDF file in Acrobat Reader.

What about the LaTeX file itself? Every LaTeX file must start with the command `\documentclass`: you can replace `article` with other formats, such as `report` (which you'll probably use to write your dissertations) and `book`. These change details of the way that LaTeX lays out the document. Similarly, every LaTeX file must have a `\begin{document}` and a `\end{document}`: between these comes all of the text which will appear in the document. The part of the file which precedes `\begin{document}` is called the *preamble*.

page 10

Typesetting Text

Exercise 1b

The `\documentclass` command can also take *optional arguments*, two of which are particularly useful. Change the opening line of your file to read

page 11

```
\documentclass[11pt,a4paper]{article}
```

The default type size is 10 point (a *point* is 1/72 of an inch): you can make it 11 or 12 point using this technique. In the US, they use paper which is smaller than A4: the `a4paper` option corrects for this.

Exercise 1c

LaTeX treats any number of consecutive spaces as a single space, and a new line as a single space. Leaving one or more blank lines starts a new paragraph. Test this out by adding to your file so that it reads

```
\documentclass[11pt,a4paper]{article}
\begin{document}
Here begins my first document written in \LaTeX. I never
   need to worry again   about hitting the   space bar
twice in succession.

Leaving a blank line starts a new paragraph.
\end{document}
```

Process (`pdfLaTeX/F6`) your file and study the results (`View pdf/F8`).

Exercise 1d

The backslash character \ is ubiquitous in LaTeX: it starts a command. For example, `\LaTeX` is the LaTeX command which prints "LaTeX". Clearly, if you want to print \ in your document, you can't just put a backslash in the LaTeX file: it would be interpreted as the start of a command. There are 9 other characters which have a special meaning: they are

```
$  &  %  #  _  {  }  ~  ^
```

All of these can be printed by preceding them with a backslash: for example, `\$` is the LaTeX command which prints a dollar sign. (However this trick doesn't work with the backslash itself.)

Add a new line to your file which outputs

> Calculate 30% of $60 & tell the chairman.

Exercise 1e

There are two types of LaTeX command names: those which consist of exactly one special character (such as `\$`), and those which consist entirely of letters (such as `\LaTeX`). Thus when LaTeX sees a backslash followed by a letter, it interprets the command name as going on until the next character which isn't a letter. In particular, you often need to end a command with a space, even when you don't actually want to print a space. For example, if you wanted to print LaTeXtonics, you couldn't write `\LaTeXtonics`, since this would be interpreted as a single command name: you have to write `\LaTeX tonics`. But then what happens if you actually want to write LaTeX tonics? The answer is that you have to use the command `\ `, (backslash space), which always prints a space: `\LaTeX\ tonics`.

Change the first sentence of your file so that it produces:

> Here begins my first LaTeX document.

Exercise 1f

To make your text **bold** or *italic* you use the commands `\textbf` or `\textit` (there are a host of other possibilities too, such as sans serif, `typewriter`, and SMALL CAPITALS). Just enclose the text you want to be affected in braces: page 110

```
\textbf{This text is bold} and this is not.
```

Incidentally, it is generally bad practice to emphasize words artificially (by making them bold or italic) in normal prose, but it is very useful in mathematics (for example, in a definition, the term being defined is commonly emphasized).

Similarly, to change the size of your font temporarily, there are commands such as `\tiny`, `\small`, `\Huge` (see Table 6.2 on page 110 for a full list). These work a bit differently: they go *inside* the braces which enclose the text whose size you want to change.

```
Here are some {\tiny tiny words} and {\large here are some large ones}.
```

To change both font and font size, simply combine the two methods:

```
\textbf{\Huge Both bold and Huge!}
```

produces: **Both bold and Huge!**

Consult page 110, and add a line to the end of your file to produce

The kitten raised its fur and *hissed* at the **Alsatian**.

Except in very special circumstances (e.g. part 2 of exercise 1n) please forget about these size-changing commands: using them will, on the whole, make your document look scrappy and ugly.

Exercise 1g

Since your document is now so long, you'd probably like it to have a title. This is easy. In the preamble (that is, after `\documentclass` but before `\begin{document}`), insert the lines

```
\title{A sample \LaTeX\ document}
\author{Your name} %Don't write ``your name'',
                   %put in your name
```

(Notice the comment: anything on a line which comes after a `%` is ignored). Now include the command `\maketitle` immediately after `\begin{document}`. Process the file and study the output.

Notice that the date (in fact, the date on which you process the file) is automatically included. If you want a fixed date to appear, you can use the `\date` command just like `\title` and `\author`: for example

```
\date{October 2008}
```

If you don't want the date to appear at all, use `\date{}`.

Change your file so that the date shown is the day you're working on this, regardless of when you actually process the file.

Exercise 1h

Most well-written documents of any length are divided into sections, subsections, and so on. LaTeX makes this easy. To start a new section called 'A very interesting section', simply type

```
\section{A very interesting section}
```

The section is automatically numbered, although you can suppress this if you want by using the *starred* version of the command:

```
\section*{A section with no number}
```

Many commands have such starred variants.

The commands `\subsection` and `\subsubsection` work similarly: if your document class is `report` or `book`, you can additionally use `\part` and `\chapter`. page 35

Modify your file by dividing it into two sections, the first called 'Some Text', and the second, starting at the current end of the file, called 'Some Maths'. Divide the first section into two subsections, 'Spaces and Paragraphs' and 'Font Changing'. (**NOTE: Be sure to use the commands `\section` and `\subsection` to do this — don't try to use `\large` etc.**). Process the document and study the result. Now add the command

```
\tableofcontents
```

immediately after `\maketitle`. Of course, in a document of this length it doesn't look so great. (If the table of contents doesn't appear, try processing the file again.)

Finally, change the document class from `article` to `report` and process the file again. Notice all of the changes, and in particular that the section numbers in the table of contents are wrong. Process the file a second time, and they should come right. What do you think has happened? This will be explained in Sheet 2. Add a `\pagebreak` command after the `\tableofcontents`. That looks better, but now the page numbers in the table of contents are wrong. Process the file again to correct them. Now that this is a report, your top level sections should be chapters. Add a line after the `\pagebreak` to start a new chapter called 'Exercise Sheet One' (use the `\chapter` command). Again, you'll need to process the file twice before the table of contents is correct.

Typesetting Mathematics

Maths can appear *inline*, as $x^2 + y^2 = z^2$, or be *displayed*, as

$$x^2 + y^2 = z^2.$$

Deciding when each of the two modes should be used is a matter of taste and experience: on the whole, beginners tend to display formulae more often than they should. Inline maths is placed between `\(` and `\)`, while displayed maths uses `\[` and `\]`. Thus the two formulae above were typeset as `\(x^2+y^2=z^2\)` and `\[x^2+y^2=z^2\]`. In fact, you can use `$` signs instead of both `\(` and `\)` to produce inline formulae: `$x^2+y^2=z^2$`. What do you think are the advantages and disadvantages of doing this?

It's very important to understand that LaTeX behaves completely differently in *maths mode* (everything between `\(` or `\[` and `\)` or `\]` is maths mode) and in *text mode*. In maths mode, letters and punctuation marks are treated by default as mathematical symbols, which are typeset in a quite different way from the same letters and punctuation marks in normal text. It's worth having a more careful look at the sample LaTeX file I handed out to see how `$`s are used to switch into and out of math mode.

Exercise 1i

Notice the use of `^` to obtain a superscript: `_` similarly yields a subscript. By default, only the single character following `^` or `_` is raised or lowered: to raise or lower more, you have to create a group. Thus, to produce $x_{i,j}$, you should type `$x_{i,j}$`: if you typed `$x_i,j$` you'd get x_i, j. Subscripts and superscripts can be mixed (thus both `$x_i^2$` and `$x^2_i$` yield x_i^2), and nested (thus `$e^{-x_i^2}$` yields $e^{-x_i^2}$).

Add to the end of your file to produce the lines:

> If x, y, z, and n are positive integers and
>
> $$x^n + y^n = z^n,$$
>
> then n is either 1 or 2.

(Be careful with the comma at the end of the displayed formula, and with the spacing in 'If x, y, z, and n': each of the four symbols has to be enclosed in its own pair of dollars. As a general rule, if you want your documents to look professional, then you should take care over punctuation after formulae, particularly displayed ones. A sentence is still a sentence even if it contains maths, and you should put full stops, commas, or no punctuation at all after formulae in just the same way as you would in normal English.)

Exercise 1j

There is an enormous range of commands producing mathematical symbols: you should refer constantly to pages 63 – 70 until you've learned the ones you use most often, which shouldn't be so many (by the way, the *AMS* symbols in tables 3.12 to 3.19 are only available if you load a relevant *package*: more on this in Sheet 3). Notice in particular the commands to produce Greek letters (`$\pi$` produces π, while `$\Pi$` produces Π). You'll use two more commands in this exercise and the next: `\sqrt` produces a square root sign around the group that follows it: so `\[\sqrt{x^2+y^2}\]` yields

$$\sqrt{x^2 + y^2};$$

and the right-pointing arrow `\to` is used both for functions ($f\colon X \to Y$) and for limits ($n \to \infty$). These two formulae are typed `$f\colon X\to Y$` and `$n\to\infty$`.

Add another line to your file which produces the displayed equation:

$$n! \sim n^n e^{-n} \sqrt{2\pi n}$$

(you can find $\sim$ in Table 3.3).

Exercise 1k

Sometimes you want to include normal text in a displayed formula. One way to do this is with `\mbox`: thus

```
\[f(x)=g(x) \mbox{ provided }x>0.\]
```

produces

$$f(x) = g(x) \text{ provided } x > 0.$$

Notice that spaces are ignored in math mode, so you have to provide your own within the mbox. Just typing `\[f(x)=g(x) provided x>0.\]` produces

$$f(x) = g(x)providedx > 0.$$

(that is, the letters of 'provided' are treated as mathematical symbols), which is not what you want.

Change the formula from the previous exercise so it now reads

$$n! \sim n^n e^{-n}\sqrt{2\pi n} \text{ as } n \to \infty$$

You'll see a slightly better way of putting text in formulae in sheet 3, in the section on packages; and another use of `\mbox` is discussed at the end of sheet 2.

Exercise 1l

Fractions are produced using `\frac`, which takes two arguments, the numerator and the denominator of the fraction respectively. Thus `\[\frac{1-x}{1+x}\]` yields

$$\frac{1-x}{1+x}.$$

Be cautious about using `\frac` in inline mode: in many instances, 1/2 looks better than $\frac{1}{2}$.

Change the formula from the previous exercise to

$$n! \sim n^{n+\frac{1}{2}} e^{-n}\sqrt{2\pi} \text{ as } n \to \infty$$

and then add another displayed equation reading:

$$\frac{n!}{n^{n+\frac{1}{2}} e^{-n}\sqrt{2\pi}} \to 1 \text{ as } n \to \infty.$$

Exercise 1m

In a formula like $\log(x+y)$, the 'log' is a single word which is set in roman type. Just typing `$log(x+y)$` treats `l`, `o`, and `g` as mathematical symbols, and hence produces $log(x+y)$, which doesn't look good. So LaTeX provides commands to typeset log and a host of other 'log-like functions' such as sin, lim, and det. The command names are just the function names with a backslash: thus page 54

```
\[\lim_{n\to\infty} f(n)\log n = \arcsin(\pi/5).\]
```

produces

$$\lim_{n\to\infty} f(n) \log n = \arcsin(\pi/5).$$

Notice that LaTeX is intelligent enough to interpret the subscript on `\lim` the way you meant it.

Add a displayed equation to the end of your file which produces:

$$\exp(i\theta) = \cos\theta + i\sin\theta.$$

Exercise 1n

A number of mathematical operators (see Table 3.5 on page 65) can take upper and lower limits: these are specified as subscripts and superscripts in the usual way. For example `$\sum_{i=1}^n x_i$` yields $\sum_{i=1}^{n} x_i$, while

```
\[\sum_{n=1}^\infty\frac{1}{n^2}=\frac{\pi^2}{6}\]
```

gives

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

(notice the different formats of the results for inline and display modes). You could also type `\[\int_0^\infty e^{-x^2} dx\]` to give

$$\int_0^{\infty} e^{-x^2} dx,$$

or `\[\bigcup_{i\in I} U_i\]` for

$$\bigcup_{i\in I} U_i.$$

Try to get LaTeX to produce the following formulae at the end of your file:

1.

$$W_\lambda^\alpha \to W_\lambda^\alpha + \delta W_\lambda^\alpha \equiv W_\lambda^\alpha + D_\lambda^{\alpha\beta}\omega^\beta.$$

(you'll find $\equiv$ in Table 3.3).

2.

$$\int_C f(z)dz = 2\pi i \sum_{\text{poles } z} R_z.$$

(pay attention to the size of the letters in the word 'poles'; and make sure that this word is typeset as text, not as mathematics).

3.

$$a_n = \frac{1}{2\pi}\int_0^{2\pi} \frac{f(z_0 + re^{i\theta})}{r^{n+1}e^{i(n+1)\theta}} ire^{i\theta} d\theta.$$

4.

$$1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cdots}}} = \frac{1+\sqrt{5}}{2}.$$

(use `\cdots` to get the $\cdots$).

Submission

My email address is `t.hall@liv.ac.uk`. Send me an email, attaching the file `yourname1.tex`.