

# Stringing words together

(with  $\text{\LaTeX}$ )

Andrew Mathas

An essay submitted in partial fulfillment of  
the requirements for the degree of  
something cool

Pure Mathematics  
University of Sydney



January 2010



## CONTENTS

<b>Introduction</b> .....	<b>iv</b>
<b>Chapter 1. Starting out</b> .....	<b>1</b>
1.1. The structure of a $\LaTeX$ document .....	1
1.2. Taking command .....	2
1.3. $\TeX$ ing the document .....	3
1.4. How to cope with really long section (or chapter) titles that go on and on and on and on and never seem to stop .....	5
<b>Chapter 2. Typing mathematics</b> .....	<b>6</b>
2.1. Mathematical alignment .....	6
2.2. Intertext .....	8
2.3. Bold statements .....	9
2.4. Cross referencing .....	10
2.5. Commutative diagrams and pictures .....	10
2.6. Tying things together .....	11
2.7. Extra p's and q's .....	11
<b>Appendix A. Including files</b> .....	<b>12</b>
<b>Appendix B. Class file options</b> .....	<b>14</b>
B1. What to do if you're not pure .....	14
B2. Staying in the past .....	14
B3. Changing times .....	14
B4. Indoctrination .....	14
<b>References</b> .....	<b>15</b>

## Introduction

The aim of this document is to tell you all you need to know to start using  $\LaTeX$  to write your fourth year essay. I'll give you the basic outline of the  $\LaTeX$  file, a run down of the commands, and a very brief introduction to writing mathematics using  $\LaTeX$ ; at the same time, show off some of the features of the  $\LaTeX$  class file that I have written for writing fourth year essays (and indeed, theses in general).

In writing this introduction to  $\LaTeX$  I've tried to illustrate (by example) how  $\LaTeX$  copes with many of the basic problems you'll encounter when you start to write your essay. This file is not intended to be a good example of how to set out a document; for example, there are gratuitous examples of subsections and so on.

I'm not going to tell you everything that you would like to know about using  $\LaTeX$ ; however, by the end of this document you will know enough to start typing — and where to look to find more information. I also recommend looking at the source for this document to see how some things are done (you can get it from the fourth year web page).

Typesetting mathematics is second in difficulty only to typesetting music. Most mathematical papers and books are written using  $\LaTeX$ , or  $\TeX$ . The  $\TeX$  program was written by Donald Knuth specifically for typesetting mathematics;  $\LaTeX$  is built on top of  $\TeX$  with the aim of taking care of most of the typesetting details for you so that you can concentrate on the content of whatever you are writing.

In my experience none of the commercial word processing programs can really cope with complex mathematical expressions such as

$$(-1)^{a_{rL}} q^{b_{rL}} \frac{\prod_{1 \leq s < t \leq r} (Q_s - Q_t)^L \cdot \prod_{1 \leq s, t \leq r} \prod_{\alpha_s \in B_s} \prod_{1 \leq k \leq \alpha_s} (q^k Q_s - Q_t)}{(q-1)^n (Q_1 \dots Q_r)^n \prod_{1 \leq s \leq t \leq r} \prod_{\substack{(\alpha_s, \alpha_t) \in B_s \times B_t \\ \alpha_s > \alpha_t \text{ if } s=t}} (q^{\alpha_s} Q_s - q^{\alpha_t} Q_t)}.$$

(As with all of the examples in this document, this comes from a real live mathematics paper.) This said, a friend of mine tells me that programs such as microsoft word are now very good at typesetting mathematics.

You are welcome to type your fourth year essay using anything you like; the School only stipulates that your essay must be typed. I would recommend using  $\LaTeX$ , but the choice is yours.

$\LaTeX$  (or more accurately,  $\TeX$ ), is actually an incredibly powerful (but low level) programming language which gives you complete control over the look and feel of your document. As a consequence, you can make it do virtually anything you want from adjusting page margins to having it compute and print  $\pi$  correct to 50 decimal places (this would be a silly thing to do, but you could do it if you wished).

The downside to all this flexibility is that  $\LaTeX$  takes a bit of effort to learn and, when you're typing it, it does not look nearly as nice on the screen as something like word; however, the output is better than that produced by other programs (I think). In particular,  $\LaTeX$  is not a wysiwyg language: that is, you don't automatically see on the screen what you type into a  $\LaTeX$  document (like, for example, in word). Rather,  $\LaTeX$  is an *embedded formatting language*; that is, you type commands into a file (using an editor), then process the file using  $\LaTeX$  and then — finally — you can see what it looks like using either `xdvi` or by printing the file. For example, to produce the symbol  $\alpha$  you actually type `\alpha` into the  $\LaTeX$  file.

As I said at the start, this introduction is not intended to be a complete guide to using  $\TeX$  and  $\LaTeX$ ; for this you should consult the many books on  $\LaTeX$  in the library such as [7] or [5]. Of these I think that Lamport's book [7] is probably the most useful. In addition, see the various short guides available from the fourth year web page. The really enthusiastic can also read all about  $\TeX$  in [6].

Finally, feel free to ask me any questions you might have about  $\LaTeX$ . Also, if there is something that you think the `usydthesis` package should do but doesn't, or if you have any suggestions improvements, let me know.

Andrew Mathas  
May 2002

## CHAPTER 1

### Starting out

#### 1.1. The structure of a L<sup>A</sup>T<sub>E</sub>X document

The basic layout of the L<sup>A</sup>T<sub>E</sub>X file for your essay will be something like the following:

```
\documentclass[BSc]{usydthesis}

\author{Your name}
\title{Your essay title}

\numberwithin{equation}{chapter}
\newtheorem{Definition}[equation]{Definition}
\newtheorem{Theorem}[equation]{Theorem}
\newtheorem{Proposition}[equation]{Proposition}
\newtheorem{Lemma}[equation]{Lemma}
\newtheorem{Corollary}[equation]{Corollary}

\theoremstyle{remark}
\newtheorem{Remark}[equation]{Remark}
\newtheorem{Example}[equation]{Example}

\begin{document}      % start of the "text"

\maketitle            % the title page
\tableofcontents      % the table of contents

\chapter{Introduction} % chapter 1!
...
\end{document}
```

All of this (and a bit more), is contained in the file `skeleton.tex` which you can download from the web page

[www.maths.usyd.edu.au/u/mathas/courses/pm4/](http://www.maths.usyd.edu.au/u/mathas/courses/pm4/).

You can probably guess what most of the code above does. The first line is not obvious: it tells L<sup>A</sup>T<sub>E</sub>X to load the class file `usydthesis` which

looks after such things as the title page for the essay. The BSc says that you are doing a Bachelor of science. The `\newtheorem` commands define *environments* for stating theorems, propositions and so on; more on this later.

As a general rule, anything in T<sub>E</sub>X which begins with a `\` is a command. Braces `{...}` are used to group things; for example to put “a word” into italics you would type `\textit{a word}`. Percentage signs `%` are comments: anything after them on the same line is ignored. (Use `\backslash`, `\{` and `\%` to type `\`, `{` and `%` respectively.)

## 1.2. Taking command

At the top of your essay (before the `\begin{document}`), you will probably want to define some *macros*. For example, you might include the following lines:

```
% natural numbers, real numbers
\newcommand{\N}{\mathbb{N}}
\newcommand{\R}{\mathbb{R}}

% macros for End(X) and Hom(X,Y)
\DeclareMathOperator{\End}{End}
\DeclareMathOperator{\Hom}{Hom}

% for f:X -> Y - the default spacing isn't great
\newcommand{\map}[2]{%
  {\ , { : } \ , #1 \! \! \longrightarrow \! #2}}
```

Let me explain these. First, unsurprisingly, `\newcommand` is used to define a new L<sup>A</sup>T<sub>E</sub>X command; the new commands above are called `\N`, `\R`, `\End` and so on. In the first two macros, `\mathbb` changes the font into the *blackboard font* so that `\N` produces  $\mathbb{N}$  — the dollar signs put L<sup>A</sup>T<sub>E</sub>X into *math<sup>1</sup> mode* which is necessary because the blackboard fonts only work inside equations. Similarly, `\DeclareMathOperator` is used to define mathematical operators; so `\End(X)` and `\End_A(X)` produce  $\text{End}(X)$  and  $\text{End}_A(X)$ , respectively.

The last macro is more complicated because it takes two *arguments*; this accounts for the `[2]` in its definition. Given this macro you can type `\f\map XY` to produce  $f: X \rightarrow Y$ ; or, for a more complicated example, `\varphi\map{\End_A(X)}{\End_A(Y)}` to produce

$$\varphi: \text{End}_A(X) \rightarrow \text{End}_A(Y).$$

---

<sup>1</sup>T<sub>E</sub>X was written by an American

(The *double dollar signs* tell T<sub>E</sub>X to display the equation, centred on a line by itself.) The *braces* are necessary in the second example because we want `\End_A(X)` and `\End_A(Y)` each to be treated as a single argument by `\map`. In the first use of `\map` we could have written `$f\map{X}{Y}$`; however, the braces are not necessary here so I omitted them.

In the `usydthesis` class file there is also a `\set` macro for making sets. Its definition is complicated so I am not going to explain it; however, as it is in the class file you can use it. For example, you can type `$\set{x\in\mathbb{R}|x^2=1}$` to produce  $\{x \in \mathbb{R} \mid x^2 = 1\}$ .

I'm actually departing a little from standard L<sup>A</sup>T<sub>E</sub>Xese above because L<sup>A</sup>T<sub>E</sub>X tries to encourage you to type `\(...\)` and `\[...\]` for in-text and displayed mathematics respectively; I don't like this and use the T<sub>E</sub>X primitives `$. . . $` and `$$ . . . $$` instead.

Finally, I should remark that there is a `\renewcommand` for redefining commands which already exist; this is necessary because `\newcommand` gives an error if the command already exists. If this happens, I suggest that you call your command something else.

### 1.3. T<sub>E</sub>Xing the document

For the impatient the basic run down is that you:

1. Edit the file; say it's called `essay.tex`.
2. Type `latex essay` to L<sup>A</sup>T<sub>E</sub>X it.
3. Type `xdvi essay` to preview it on the screen.
4. Type `dvips essay` to print it.

Here, and below, I'm assuming that you are using UNIX. In particular, when I say *Type* I mean *type at the UNIX shell prompt*.

In order to use T<sub>E</sub>X you must first create a text file which contains what you want to typeset (i.e. your essay). There are various editors available on the School computers; of these the easiest to use is probably `nedit`. In addition, you will also find `vi` and `emacs` and their extensions `gvim` and `xemacs` respectively.

In order to use the `usydthesis` class file L<sup>A</sup>T<sub>E</sub>X must be able to find the file `usydthesis.cls`. This will happen automatically on the linux machines within the School. From outside the school you will have to download the class file from the web page

`www.maths.usyd.edu.au/mathas/courses/pm4/`.

If you put a copy of this file onto the directory containing your essay everything should work fine. (You can borrow an installation CD from Robert Pearson if you are interested in putting T<sub>E</sub>X onto your computer; if you are using Linux then T<sub>E</sub>X should already be installed.)

When you first start working with  $\LaTeX$  you are often going to find that `essay.tex` contains errors. The most common errors are due to *missing braces* or *missing dollar signs*; these must always occur in matching pairs. As with all programs,  $\LaTeX$ 's error messages are sometimes informative; however, most of the time they tend to be slightly cryptic and the errors can take some time to track down and fix. When in doubt, just press `r` (for “run”) and  $\LaTeX$  will skip over all of the errors (if it can) and then you can look at the file using `xdvi` which will often help you spot the mistake.

The best strategy is to  $\LaTeX$  the file often so that not so much changes between  $\TeX$ ing; this helps you isolate the errors. After a bit of practise you'll find that most of the time the document  $\TeX$ s first time.

Finally, in order to save paper, you shouldn't print the file too often.

**1.3.1. System caveats.** The following comments only apply if you are using UNIX. If you have a PC then you may have an editor which combines the `edit-latex-xdvi` cycle. Actually, it is also possible to do this using UNIX editors; but I've never thought it worth the trouble.

1.3.1.1. *xdvi*. Rather than typing “`xdvi essay`” as above you are much better off typing “`xdvi essay &`” as this way `xdvi` runs in background; as a consequence, you will still be able to type commands into the UNIX shell window. Better still, put the line

```
alias xdvi 'xdvi \!* &'
```

into your `.cshrc` file and you can just type “`xdvi essay`”, but with this added functionality. Actually, you can do better than this; I have `xdvi` aliased to something like

```
alias xdvi 'xdvi -s 6 -geo 540x890-0+0 \!* &'
```

For a complete description of the options to `xdvi` see the UNIX manual page (i.e. type `man xdvi`).

1.3.1.2. *dvips*. All of the options to `dvips` are again described in the manual. If you type `dvips --help` you'll see a compact listing of all of these options. The most useful option is `-pp`; for example, if you type

```
dvips -pp 3-5,16 essay
```

then `dvips` will only print only pages 3–5 and page 16 of `essay.tex`. Another useful variant is the alias

```
alias dvim 'dvips -f \!* | mpage -2'
```

which will make `dvim essay` print your essay with two pages to a page.

### **1.4. How to cope with really long section (or chapter) titles that go on and on and on and on and never seem to stop**

There is no real problem in using long titles. The only thing that you need to keep in mind is that such titles will be too long for the running headers at the top of the page. To get around this the commands `\chapter`, `\section...` all take an optional argument which defines a shorter title to be used by the headings. The syntax is

```
\chapter[short title]{Long title}.
```

For example, the running header on this page is “Long titles” rather than the long title of the section.

(The running header on even numbered pages is the title of the chapter and on odd numbered pages it is the title of the section.)

## CHAPTER 2

### Typing mathematics

I'm not going to give an in depth tutorial on how to type mathematics using  $\text{\TeX}$ ; for this see [7, 6, 5]. The basics are that:

- Mathematics that appears in the middle of text is typed in between dollar signs:  $\$ \dots \$$  — or  $\backslash (\dots \backslash)$ .
- Equations that you want displayed on a separate line are typed in between double dollar signs:  $\$\$ \dots \$\$$  — or  $\backslash [\dots \backslash]$ .
- Groups of symbols etc. are treated as a single entity by enclosing them in between *matching* braces:  $\{ \dots \}$ .
- Subscripts are done using an underscore  $\_$  and superscripts using a caret  $\^$ ; for example, to write  $\sum_{i=1}^n$  you type  $\$\backslash \text{sum}_{\{i=1\}}^n \$$ . Note that the subscript  $i = 1$  is surrounded by matching braces; you could also write  $\^{\{n\}}$  rather than  $\^n$  but there is no need.
- Mathematics symbols have names like  $\backslash \text{alpha} = \alpha$ ,  $\backslash \text{oplus} = \oplus$ ,  $\backslash \text{dots} = \dots$  and so on. For a reasonably complete list see the “list of  $\text{\TeX}$  symbols” link on the fourth year web page.

The next few sections describe how to align equations in  $\text{\LaTeX}$ ; actually, I'm going to describe various commands provided by some A.M.S.<sup>1</sup> packages; these are loaded for you by the `usydthesis` class file.

#### 2.1. Mathematical alignment

One way to align equations is to use the `align` environment. *Environments* are  $\text{\LaTeX}$ 's way of grouping large blocks of text; they always take the form `\begin{align} \dots \end{align}`.

For example if you wanted to write

$$\begin{aligned} (v, v) &= (a\alpha_r + b\alpha_s, a\alpha_r + b\alpha_s) = a^2 - 2ab \cos\left(\frac{\pi}{m}\right) + b^2 \\ &= \begin{cases} (a - b \cos(\frac{\pi}{m}))^2 + b^2 \sin^2(\frac{\pi}{m}), & \text{if } m < \infty, \\ (a - b)^2, & \text{otherwise.} \end{cases} \end{aligned}$$

then you could type

---

<sup>1</sup>American Mathematical Society

```

\begin{align*}
(v, v) &= (a\alpha_r + b\alpha_s, a\alpha_r + b\alpha_s) \\
&= a^2 - 2ab\cos(\frac{\pi}{m}) + b^2 \\
&= \begin{cases}
\big(a - b\cos(\frac{\pi}{m})\big)^2 \\
+ b^2\sin^2(\frac{\pi}{m}), & \text{if } m < \infty, \\
(a - b)^2, & \text{otherwise.}
\end{cases} \\
\end{align*}

```

A few comments. First, `&` is used to mark where you want the equations aligned and `\\` is used to mark the end of a line.

Next, I have actually used the `align*` environment here and not the `align` environment; without the star these equations (each line of them in fact) would also be given an equation number. As a general rule, in  $\LaTeX$  the star forms of commands suppress (equation) numbers. For a second example, see the `\chapter*{Introduction}` in the introduction to this document.

Next, if you just type *text* in mathematics it doesn't come out the way you would like it to; compare “*if*  $m < \infty$ ” with “if  $m < \infty$  which was produced with `\text{if } m < \infty`”. The way to insert text into equations is to use the `\text` command; this command is also provided by the A.M.S. packages. Note that above we have `\text{if }`; the `\_` in the `\text` is to put a space between the “if” and what follows after it. The `\text` command is also intelligent enough to use the correct font.

Another command in the example above is `\frac`; as you see this produced fractions. The general usage is `\frac{A}{B}` where, as usual, the braces can be omitted when “A” or “B” is only a single character or command. Another interesting command is `\big` which makes the following bracket bigger. There are also `\bigg`, `\Big` and `\Bigg` variants; each of these can be applied to any delimiters: for example, to `(, )`, `[, ]`, `\{` and `\}`.

Finally, the `cases` environment is also used above. Notice that `cases` environment also aligns formulae and that the ampersands in the `cases` environment do not affect the outer `align*` environment.

Similarly, there is also an `alignat` environment which is useful for putting comments after your equations. For example, to produce

$$(T_s - q)(T_s + 1) = 0, \quad \text{for } s \in S, \quad (2.1)$$

$$(T_r T_s \dots)_{m_{rs}} = (T_s T_r \dots)_{m_{rs}}, \quad \text{for } r \neq s \in S, \quad (2.2)$$

you could type

```
\begin{xalignat}{2}
(T_s-q)(T_{s+1})&=0, &\&\text{for\ } s\in S, \\
(T_rT_s\dotscdots)_{m_{rs}}&=(T_sT_r\dotscdots)_{m_{rs}}, \\
&\&\text{for\ } r\neq s\in S, \\
\end{xalignat}
```

This time I have used the `alignat` environment rather than its `*`-form; hence the equation numbers. For a description of these two environments see the “Introduction to AMSLaTeX” on the web page.

## 2.2. Intertext

You can actually accomplish the alignments above using L<sup>A</sup>T<sub>E</sub>X’s `arrays`; however, there is a feature (which I like a lot) in the A.M.S. alignment environments which allows you to insert lines of text in between equations without losing the alignment. This is done with the `\intertext` command. For example, to achieve

$$f_{us}T_i = \begin{cases} \frac{(q-1)\text{res}_t(i)}{\text{res}_t(i)-\text{res}_s(i)}f_{us} + f_{ut}, & \text{if } s \triangleright t, \\ \frac{(q-1)\text{res}_t(i)}{\text{res}_t(i)-\text{res}_s(i)}f_{us} + \frac{(q\text{res}_s(i)-\text{res}_t(i))(\text{res}_s(i)-q\text{res}_t(i))}{(\text{res}_t(i)-\text{res}_s(i))^2}f_{ut}, & \text{if } t \triangleright s. \end{cases}$$

*If  $t$  is not standard then*

$$f_{us}T_i = \begin{cases} qf_{us}, & \text{if } i \text{ and } i+1 \text{ are in the same row of } s, \\ -f_{us}, & \text{if } i \text{ and } i+1 \text{ are in the same column of } s. \end{cases}$$

you would type the following

```
\begin{align*}
f_{us}T_i&=\begin{cases}
\frac{(q-1)\text{res}_t(i)}{\text{res}_t(i)-\text{res}_s(i)}f_{us} \\
+f_{ut}, & \text{\text{if\ } s\unrhd t, \ll[5pt]} \\
\frac{(q-1)\text{res}_t(i)}{\text{res}_t(i)-\text{res}_s(i)}f_{us} \\
+\frac{(q\text{res}_s(i)-\text{res}_t(i))(\text{res}_s(i)-q\text{res}_t(i))}{(\text{res}_t(i)-\text{res}_s(i))^2}f_{ut}, & \text{\text{if\ } } \\
t\unrhd s. \\
\end{cases} \\
\intertext{If }t$ is not standard then}
f_{us}T_i&=\begin{cases}
qf_{us}, & \text{\text{if } $i$ and } $i+1$ are \\
& \text{in the same row of } $s$, \ll} \\
-f_{us}, & \text{\text{if } $i$ and } $i+1$ are \\
& \text{in the same column of } $s$.} \\
\end{cases} \\
\end{align*}
```

Note that the two equals signs match up and that line in between the two equations (which is typed inside the `\intertext{...}`), is flush with the left margin. I agree that this isn't pretty, but that's how it's done.

### 2.3. Bold statements

The file `skeleton.tex` contains a number of lines of the form

```
\newtheorem{Theorem}[equation]{Theorem}.
```

For a description of what this is really doing again see the `AMSLATEX` introduction on the web page. These commands all create “theorem like” environments. For example,

#### 2.3. Theorem (Fermat with some help from Wiles).

*Suppose that  $n$  is a positive integer such that  $a^n + b^n = c^n$  for some positive integers  $a, b$  and  $c$ . Then either  $n = 1$  or  $n = 2$ .*

**Proof.** Damn, I should have made the margins bigger. □

was produced by the `LATEX` source

```
\begin{Theorem}
[Fermat with some help from Wiles]
\leavevmode\newline
\label{Fermat}
Suppose that  $n$  is a positive integer such that
 $a^n + b^n = c^n$  for some positive integers  $a, b$ 
and  $c$ . Then either  $n = 1$  or  $n = 2$ .
\end{Theorem}

\begin{proof}
Dam, I should have made the margins bigger.
\end{proof}
```

A couple of points. The “Fermat with some help from Wiles” is an optional argument to the theorem environment; you can leave it out. The proof environment produces both the initial “Proof” and the square which ends the proof. It took me quite a while to work out how to make `LATEX` start a new line after the Wiles (most things you try give an error message). This trick is accomplished by `\leavevmode\newline`.

For the meaning of `\label{Fermat}` see section 2.4.

Finally, for definitions I recommend using sans serif fonts; these are made with `\textsf{sans serif fonts}`.

## 2.4. Cross referencing

$\LaTeX$  provides `\ref` and `\label` commands for cross referencing. To refer to Fermat’s Theorem 2.3 above one simply types `\ref{Fermat}`. This works because of the `\label{Fermat}` inside the theorem environment. In a similar way I produced the “section 2.4” at the end of section 2.3. When you insert a new label you must  $\LaTeX$  the document twice before it will have any effect. (Note: `\ref{...}` does *not* print “Theorem 2.3”; only 2.3!)

There is a similar mechanism for citing references. At the end of this file you will find a bibliography section.  $\LaTeX$  has an automatic way of citing references. For example, to refer to the  $\TeX$ book in the bibliography I type `\cite{TeX}`. The *key* used by the command `\cite` is the first argument of the corresponding `\bibitem` command in the bibliography. For more details see [7]. The more adventurous might also want to look at  $\text{Bib}\TeX$ .

## 2.5. Commutative diagrams and pictures

The easiest way that I know of to do (general) commutative diagrams or pictures is to use the `pstricks` package. To learn how to use this you’ll have to read the documentation which can be found in

```
/usr/local/texlive6/texmf/source/generic/pstricks/doc/.
```

To use `pstricks` you need the line `\usepackage{pstricks}` at the top of your document; some parts of `pstricks` also require you to load other packages (see the documentation). I’ll give one example. The commutative diagram

## 2.6. Tying things together

$\TeX$  occasionally breaks lines so that a single (mathematical) character begins a new line; this doesn’t look so good. You can force  $\TeX$  to tie two characters together using `~`. For example, if you want to ensure that a line break does not occur in the middle of “an  $X$ ” you would type `an~$X$`.

## 2.7. Extra p’s and q’s

When we mark your essay, in addition to the mathematics in your essay being correct we would also like to read well constructed and grammatically correct sentences! Correct spelling and punctuation throughout are also much appreciated by the markers. Proof reading is a hard job, so there are bound to be some typos; however, spelling mistakes shouldn’t appear in your essay because there are various programs available for checking

spelling. For example, try `ispell -t essay.tex`; this program ignores some (but unfortunately not all)  $\LaTeX$  commands.

You can also try `~mathas/bin/same essay.tex`; this will find any repeated words in your document.

## APPENDIX A

### Including files

As fourth year essays are not very long (usually around 40–60 pages), you can get away with putting the whole document into a single file; however, you may find it easier to break the document up into separate files; for example, having one file for each chapter.

The easiest way to do this is to have a *main* file, which is essentially the skeleton file we started with, together with any macros you define and any other general formatting commands. Let's suppose that the main file is still called `essay.tex` and that in addition we also have `intro.tex`, `chapt1.tex` and so on for the chapters. The file `essay.tex` will look something like the following:

```
\documentclass[BSc]{usydthesis}

\author{Your Name}
\title{Your title}

% theorem definitions, macros etc
....

\begin{document}      % start of the "text"
\maketitle           % creates the title page
\tableofcontents     % table of contents

\pagenumbering{roman}
\include{intro}

\newpage
\setcounter{page}{1}\pagenumbering{arabic}
\include{chapt1}
...

\end{document}
```

The chapters are included using `\include{...}`. Each of the chapter files would start very simply; for example, `intro.tex` might begin as follows:

```
\chapter*{Introduction}
Typesetting mathematics is second in difficulty
only to typesetting music. Most mathematical
papers and books are written using \LaTeX, an
extension of \TeX, which is a program which was
written by Donald Knuth specifically for
typesetting mathematics.
```

To  $\LaTeX$  the essay you type `latex essay` as before (typing `latex intro` will generate an error message because most of the formatting information for `intro.tex` is contained in `essay.tex`).  $\LaTeX$  will find all of the chapters and also keep track of things like page and chapter numbering for you automatically this way.

Finally, there is also an `\includeonly` command which tells  $\LaTeX$  to only include certain chapters. For example, if you put

```
\includeonly{chapt2}
```

at the top of `essay.tex` then  $\LaTeX$  would only process the contents of chapter 2; however,  $\LaTeX$  still remembers the page numbers and the referencing information for the other chapters. This is very useful if your document is quite large and you are only working on a small piece of it.

## APPENDIX B

### Class file options

#### B1. What to do if you're not pure

By default the `usydthesis` class file assumes that you are a student in Pure Mathematics. You can also use it if you are doing a Bachelor of arts. For example, if you're in applied (doing a bachelor of arts) or statistics (doing a bachelor of science) then you can tell the package this by modifying the `documentclass` statement at the top of the  $\TeX$  file as follows:

```
\documentclass[BA,applied]{usydthesis}
\documentclass[BSc,stats]{usydthesis}
```

#### B2. Staying in the past

In a similar way, if you want to set the “publication” date of your essay (see the title page) then you should redefine `\Today` before `\maketitle` is called in the preamble. For example, you might want

```
\renewcommand{\Today}{October 1342}
```

#### B3. Changing times

By default `usydthesis` uses the *times* fonts as I think that these look a little better than the *computer modern fonts* which come with  $\TeX$ . There is a `notimes` option to the class file just in case you don't want to use these fonts. For example, if you are in statistics you would call `usydthesis` with the following line.

```
\documentclass[BA,stats,notimes]{usydthesis}
```

#### B4. Indoctrination

The class file also allows for masters and Ph.D.'s. For example, .

```
\documentclass[MSc]{usydthesis}
\documentclass[PhD]{usydthesis}
```

## References

- [1] K. BREMKE AND G. MALLE, *Reduced words and a length function for  $G(e, 1, n)$* , *Indag. Math.*, **8** (1997), 453–469.
- [2] C. W. CURTIS AND I. REINER, *Representation Theory of Finite Groups and Associative Algebras*, Interscience, New York, 1962.
- [3] R. DIPPER AND G. JAMES, *The  $q$ -Schur algebra*, *Proc. L.M.S.* (3), **59** (1989), 23–50.
- [4] ———,  *$q$ -Tensor space and  $q$ -Weyl modules*, *Trans. A.M.S.*, **327** (1991), 251–282.
- [5] M. GOOSSENS ET. AL, *The Latex companion*, Addison-Wesley, USA, 1994.
- [6] D. E. KNUTH, *The T<sub>E</sub>XBook*, Addison-Wesley, USA, 1988
- [7] L. LAMPORT, *L<sup>A</sup>T<sub>E</sub>X: A document preparation system*, Addison-Wesley, USA, 1986.