Using LaTeX to Write a PhD Thesis

Dr Nicola Talbot

School of Computing Sciences
University of East Anglia

6th October, 2004

Abstract

This document is aimed at PhD students who want to use LaTeX to typeset their PhD thesis. If you are unfamiliar with LaTeX I would recommend that you first read LaTeX for Complete Novices [5].

This document and associated files are available on-line at http://theoval.cmp.uea.ac.uk/~nlct/latex/thesis/thesis.html

Contents

Chapter 1

Introduction

Many PhD students in the sciences are being encouraged to produce their PhD thesis in LaTeX, particularly if their work involves a lot of mathematics. This document is intended as a brief guide on how to structure your document, and how to define new page styles, chapter headings and so on. If you have never used LaTeX before, I would recommend that you first read LaTeX for Complete Novices [5], as this document assumes you have a basic knowledge of LaTeX.

Throughout this document, source code is illustrated in the form:

```
This is an \textbf{example}.
```

The corresponding output is illustrated as follows:

This is an **example**.

Command definitions are shown in a typewriter font in the form:

`\documentclass[`*options*`]{`*class file*`}`

Definition

Chapter 2

Getting Started

If you have been told to use a particular class file, use that one, otherwise I would recommend that you use the `report` class file. Before you start your document, consider first what kind of structure it should have. Unless you have been told otherwise, I would recommend that you start out with a skeletal document that looks something like the following:

```
\documentclass[a4paper]{report}

\begin{document}

\title{A Sample PhD Thesis}
\author{A. N. Other}
\date{July 2004}

\maketitle

\pagenumbering{roman}
\tableofcontents
\listoffigures
\listoftables

\chapter*{Acknowledgements}

\begin{abstract}
\end{abstract}

\pagenumbering{arabic}

\chapter{Introduction}
\label{ch:intro}

\chapter{Technical Introduction}
\label{ch:techintro}

\chapter{Method}
\label{ch:method}

\chapter{Results}
\label{ch:results}

\chapter{Conclusions}
\label{ch:conc}

\bibliographystyle{plain}
\bibliography{thesis}
```

```
\end{document}
```

If you do this, it will help ensure that your document has the correct structure before you begin with the actual contents of the document.

Chapter 3

Splitting a Large Document into Several Files

Some people prefer to place each chapter of a large document in a separate file. You can do this by using the command

\include{*filename*}

Definition

If you only want to work on one or two chapters, you can tell LaTeX to only include those files using the command

\includeonly{*file list*}

Definition

in the preamble, where *file list* is a comma separated list of files you want included. LaTeX will still read in all the cross-referencing information for the missing chapters, but won't include them in the DVI file. There is a definite advantage to this if you have, say, a large number of images in your results chapter, which you don't need when you're working on, say, the technical introduction. You can still reference all the figures in the omitted chapter, as long as you have previously LaTeXed the document without the \includeonly command.

The example given in Chapter 2 can now be split into various files:

File thesis.tex:

```
\documentclass[a4paper]{report}

\begin{document}

\title{A Sample PhD Thesis}
\author{A. N. Other}
\date{July 2004}

\maketitle

\pagenumbering{roman}
\tableofcontents
\listoffigures
\listoftables

\chapter*{Acknowledgements}

\begin{abstract}
\end{abstract}

\pagenumbering{arabic}
```

```
\include{intro}

\include{techintro}

\include{method}

\include{results}

\include{conc}

\bibliographystyle{plain}
\bibliography{thesis}

\end{document}
```

File intro.tex:

```
\chapter{Introduction}
\label{ch:intro}
```

File techintro.tex:

```
\chapter{Technical Introduction}
\label{ch:techintro}
```

File method.tex:

```
\chapter{Method}
\label{ch:method}
```

File results.tex:

```
\chapter{Results}
\label{ch:results}
```

File conc.tex:

```
\chapter{Conclusions}
\label{ch:conc}
```

If you only want to work on, say, the Method and the Results chapters, you can place the following command in the preamble:

```
\includeonly{method,results}
```

Chapter 4

Changing the Document Style

It is possible to redefine \chapter, \section etc in order to change the heading style for your document. If you want to do this I recommend that you create a or class file to do this. There are two main reasons for this: firstly, some of the commands involved use an @ character which behaves differently depending on whether or not it occurs in a class/package or in a normal .tex file, and secondly, if you place all these commands in your main document, you may confuse the spell checker or word count application[1].

So, should you create a package or a class file? Packages should be designed to be independent of the class file. For example, the graphicx package works irrespective of whether you are using the report, article, slide etc class file. If the commands or environments that you want to define are somehow dependent on a particular class file, then you should create a new class file that is based on the one you want. If you are redefining chapter or section styles, then this is dependent on the overall document style, that is, it's dependent on the class file. So, you should create a new class file that modifies the existing one, rather than creating a package.

Let's have an example. If you want to create a new class called, say, mythesis, you will need to create a file called mythesis.cls, and the start of your file should look something like:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{mythesis}
```

Next you need to specify what to do with any options passed to this class file. Since we don't need to define any new options for this example, we can simply pass all options on to the report class file:

```
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{report}}
```

Once all options have been declared, they need to be processed:

```
\ProcessOptions
```

Now the report class needs to be loaded:

```
\LoadClass{report}
```

and the very last line of your file should have the command

```
\endinput
```

The contents of this new class file should be inserted between the \LoadClass{report} and \endinput commands. You will then need to modify your source code, thesis.tex, so that it uses this new class file:

```
\documentclass[a4paper]{mythesis}
```

4.1 Modifying Object Specific Text

The report class file defines various commands that produce words such as "Contents", "Chapter", "Bibliography". These commands, and their default values are listed in Table 4.1.

So, suppose you want your figures and tables to be labelled Fig. and Tab. instead of Figure and Table, then you could add the following lines to mythesis.cls:

```
\renewcommand{\figurename}{Fig.}
\renewcommand{\tablename}{Tab.}
```

[1] for information on counting the number of words in your document, see the documentation for the cmpreprt class file

Table 4.1: Default Names

\contentsname	Contents
\listfigurename	List of Figures
\listtablename	List of Tables
\bibname	Bibliography
\indexname	Index
\figurename	Figure
\tablename	Table
\partname	Part
\chaptername	Chapter
\appendixname	Appendix
\abstractname	Abstract

4.2 Changing the Section Headings

It is possible to customise the way your section, subsection etc headings appear by redefining the corresponding commands \section, \subsection etc using the command:

\@startsection{*type*}{*level*}{*indent*}{*beforeskip*}{*afterskip*}{*style*}

Definition

The six arguments are as follows:

type The sectioning type. This should be one of: section, subsection, subsubsection, paragraph or subparagraph. (Note no backslash.)

level This is the sectioning level as indicated in Table 4.2.

indent This should be a length, specifying the indentation from the left margin.

beforeskip The absolute value of the *beforeskip* specifies how much vertical distance to leave before the heading. If *beforeskip* is negative, the first paragraph following the section heading will not be indented.

afterskip The absolute value of the *afterskip* specifies how much vertical distance to leave after the heading. If *afterskip* is negative, the text following the sectioning command will appear on the same level as the section heading.

style The *style* are the declarations required to set the style of the heading (e.g. \itshape for an italic heading.)

Table 4.2: Section Levels

part	-1
chapter	0
section	1
subsection	2
subsubsection	3
paragraph	4
subparagraph	5

As an example, suppose you want to change the section headings so that they appear in a large italic font, you could do something like:

```
\renewcommand{\section}{\@startsection
{section}%                    % the name
{1}%                          % the level
```

```
{0mm}%                          % the indent
{-\baselineskip}%               % the before skip
{0.5\baselineskip}%             % the after skip
{\normalfont\large\itshape}} % the style
```

See *A Guide to LaTeX* [2] for further information.

There is a counter called `secnumdepth` that controls what level the sections have numbers. The levels correspond to those shown in Table 4.2. By default this value is 2, so only parts, chapters, sections and subsections have associated numbers. You can use `\setcounter` to change the value of `secnumdepth`. So, for example, if you want the `\paragraph` command to produce a number, do

```
\settocounter{secnumdepth}{4}
```

4.3 Changing the Chapter Headings

If you want to change the chapter or part heading style, you can't use the `\@startsection` command. Instead you should use the `\secdef` command. If you load `report.cls` into a text editor, you will see that both the `\part` and `\chapter` commands use `\secdef`. The definition of `\chapter` has the line

```
\secdef\@chapter\@schapter
```

and `\part` has the line

```
\secdef\@part\@spart
```

The first argument to `\secdef` tells LaTeX what to do if the unstarred version is used, and the second argument tells LaTeX what to do if the starred version is used. So the command

```
\chapter{Introduction}
```

will use the command `\@chapter`, whereas the command

```
\chapter*{Acknowledgements}
```

will use the command `\@schapter`. The commands `\@chapter` and `\@schapter` use the commands `\@makechapterhead` and `\@makeschapterhead`, respectively, to format the chapter heading, so if you want to change the chapter format, you will need to redefine the commands `\@makechapterhead` and `\@makeschapterhead`. The easiest way to do this is to look for the code for these commands in `report.cls` and copy them over to your new class file, `mythesis`, described earlier, and edit the appropriate formatting commands.

For example, suppose you want a line to appear above and below the chapter heading, and have the chapter heading appear in small capitals, you could do:

```
\renewcommand{\@makechapterhead}[1]{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright \normalfont
    \hrule                                          % horizontal line
    \vspace{5pt}%                                   % add vertical space
    \ifnum \c@secnumdepth >\m@ne
        \huge\scshape \@chapapp\space \thechapter % Chapter number
        \par\nobreak
        \vskip 20\p@
    \fi
    \interlinepenalty\@M
    \Huge \scshape #1\par                           % chapter title
    \vspace{5pt}%                                   % add vertical space
    \hrule                                          % horizontal rule
    \nobreak
    \vskip 40\p@
  }}

\renewcommand{\@makeschapterhead}[1]{%
```

```
\vspace*{50\p@}%
{\parindent \z@ \raggedright
  \normalfont
  \hrule                                    % horizontal line
  \vspace{5pt}%                             % add vertical space
  \interlinepenalty\@M
  \Huge \scshape #1\par                     % chapter title
  \vspace{5pt}%                             % add vertical space
  \hrule                                    % horizontal line
  \nobreak
  \vskip 40\p@
}}
```

You can download the file mythesis.cls which includes all the examples covered so far in this chapter.

4.4 Adding to the Table of Contents

Starred versions of the sectioning commands are not added to the table of contents by default, but they can be added using:

\addcontentsline{*file*}{*type*}{*text*}

Definition

file This should be the extension of the file where the contents are written. So this will be `toc` for the table of contents, `lof` for the list of figures and `lot` for the list of tables.

type This is the type of object you are adding to the contents. e.g. chapter, section, figure.

text This is the text that should go in the contents.

For example, the bibliography is generated using a starred version of the \chapter command, so it doesn't get added to the table of contents. To add it to the table of contents, you can do

```
\addcontentsline{toc}{chapter}{\bibname}
```

The counter `tocdepth` controls the section level depth in the table of contents. The levels corresponding to the sections are shown Table 4.2.

The `report` class file sets `tocdepth` to 2, which means that only the parts, chapters, sections and subsections will be entered into the table of contents. You can use \setcounter to change the value of `tocdepth`. For example, to also include the subsubsections, paragraphs and subparagraphs, do:

```
\setocounter{tocdepth}{5}
```

4.5 Defining a New Page Style

There are two page styles pre-defined by LaTeX[2]: `empty` and `plain`. These page styles can be selected either using:

\pagestyle{*style*}

Definition

to change the style "from this point onwards", or

\thispagestyle{*style*}

Definition

to change the style for a specific page.

Both these commands call the command \ps@*style*, and it is this command which redefines the header and footer. So, `\pagestyle{plain}` calls the command `\ps@plain` which in turn calls the commands that redefine the header and footer, and `\pagestyle{empty}` calls the command `\ps@empty` and so on.

[2] most of the standard class files, including `report`, also define the page styles `headings` and `myheadings`

So, to define a new page style called, say, `thesis`, you first need to define a command called `\ps@thesis`. Since the command name contains an `@` character, this definition needs to go in a style file or class file.

The header and footers for odd and even numbered pages can be specified by redefining the commands: `\@oddhead`, `\@evenhead`, `\@oddfoot` and `\@evenfoot`.

For example, suppose you want the new page style to have empty headers, and the footers to contain the page number with a dash on either side (e.g. -10-) centred, then you could do:

```
\newcommand{\ps@thesis}{
   \renewcommand{\@oddhead}{}%     header blank
   \renewcommand{\@evenhead}{}%    header blank
   \renewcommand{\@oddfoot}{\hfill-\thepage-\hfill}%
   \renewcommand{\@evenfoot}{\hfill-\thepage-\hfill}%
}
```

Note that if you are using the default `oneside` option to the `report` class file, only the `\@oddhead` and `\@oddfoot` commands will have any effect. If you want the odd and even numbered pages to be different, you must remember to use the `twoside` option[3]. It is also possible to customise page styles using the `fancyhdr` package (by Piet van Oostrum). See *A Guide to LaTeX* [2] for an example.

Unless you are told otherwise, I recommend that you use the `headings` page style.

[3] this generally isn't appropriate for a thesis

Chapter 5

Generating a Bibliography

When you're writing a large document such as a PhD thesis, I would strongly recommend that you use BibTeX rather than typing up the bibliography in a `thebibliography` environment. If you use BibTeX:

1. Only the references that you cite are included in the bibliography. (Examiners tend to fault uncited references.)
2. References are displayed in a consistent manner.
3. Entries can be sorted in order of citation or alphabetically.
4. The style can easily be changed simply by using a different bibliography style file.

You may have noticed that the example file listed in Chapter 2 had the lines:

```
\bibliographystyle{plain}
\bibliography{thesis}
```

The command

\bibliographystyle{*style*}

Definition

indicates which BibTeX style file (`.bst`) to use without the extension. The above example uses `plain.bst`. The command

\bibliography{*database*}

Definition

indicates which database (`.bib`) to use. The above example uses the database `thesis.bib`, which we will need to create. Since the document currently doesn't have any `\cite` commands, and `thesis.bib` does not yet exist, the DVI file does not yet have a bibliography.

There are many bibliography styles, but the basic ones are:

`abbrv` Entries sorted alphabetically with abbreviated first names, months and journal names.

`alpha` Entries sorted alphabetically with the citation represented by author surname and year instead of a number.

`plain` Entries sorted alphabetically, with the citation represented by a number.

`unsrt` Entries sorted according to citation with the citation represented by a number.

See *A Guide to LaTeX* [2] or *The LaTeX Companion* [1] for information about other bibliography styles, and check with your supervisor to see if there is a particular style you should be using.

Entries in the bibliography database should have the following form:

```
@entry type{keyword,
   field name = "text",

       ⋮

   field name = "text"
}
```

where *entry type* indicates the type of entry (e.g. book or article). Standard entry types are listed in Table 5.1.

Table 5.1: Standard BiBTeX entry types

`article`	Article from a journal
`book`	Published book
`booklet`	Printed work without a publisher
`conference`	Identical to `inproceedings`
`inbook`	Part, chapter, section etc of a book
`incollection`	A chapter of a book with its own author and title
`inproceedings`	An article in a conference proceedings
`manual`	Technical documentation
`mastersthesis`	A master's thesis
`misc`	Non-standard work
`phdthesis`	PhD thesis
`proceedings`	Conference proceedings
`techreport`	Report published by an institution
`unpublished`	Unpublished work with an author and title

Within an entry, *keyword* is a short label that is used to cite this work with the `\cite` command. If you have written bibliographies with the `thebibliography` environment, it's the same as the argument to `\bibitem`. There then follows a comma-separated list of fields of the form *field name* `=` *text*. The *field name* indicates what kind of field it is, e.g. `title`, `author`. Table 5.2 lists the standard fields. Note that some bibliography styles may define additional non-standard fields, such as `email` or `url`. See *A Guide to LaTeX* [2] or *The LaTeX Companion* [1] for information about other fields not listed in Table 5.2.

The required and optional fields for the standard entry types are listed in Table 5.3. If an entry has a field that is neither required nor optional, BibTeX will ignore it. This means that you can have a field called, say, `abstract`, which will be ignored by the standard bibliography styles, but will be included if you use a bibliography style that has an `abstract` field. So you can store additional information in the database which won't appear in the bibliography.

Authors should be entered in one of the following formats:

- *forenames von surname*
- *von surname*, *forenames*
- *von surname*, *jr*, *forenames*

Examples:

Entry	**Output ("abbrv" style)**
`"Alex Thomas von Neumann"`	A.T. von Neumann
`"John Chris {Smith Jones}"`	J.C. Smith Jones
`"van de Klee, Mary-Jane"`	M.-J. van de Klee
`"Smith, Jr, Fred John"`	F.J. Smith, Jr
`"Maria {\uppercase{d}e La} Cruz"`	M. De La Cruz

Compare the last example with: `"Maria De La Cruz"` which would produce: M. D. L. Cruz, which is incorrect.

Multiple authors should be separated by the keyword `and`. Here is an example using the `book` entry:

Table 5.2: Standard BiBTeX fields

address	Publisher/Institution's address
author	Author names
booktitle	Title of book where only a part of the book is being cited
chapter	Chapter or section number
edition	The edition of the book
howpublished	How a non-standard work was published
institution	The institute sponsoring the work
journal	The name of the journal
month	The month the work was published
note	Any additional information
number	The number of the journal, technical report etc
organization	Organization sponsoring conference or manual
pages	Page number or page range
publisher	Publisher's name
school	Academic institution where thesis was written
series	Name of a series
title	The title of the work
type	The type of technical report
volume	The volume number.

Table 5.3: Required and Optional Fields

Entry Type	**Required Fields**	**Optional Fields**
article	author, title, journal, year	volume, month, note, number, pages
book	author or editor, title, publisher, year	address, edition, volume or number, month, note, pages, series
booklet	title	author, address, howpublished, month, note, year
inbook	author or editor, chapter or pages, title, publisher, year	address, edition, volume or number, month, note, series, type
incollection	author, title, booktitle, publisher, year	address, chapter, editor, edition, volume or number, month, note, pages, series, type
inproceedings	author, title, booktitle, year	address, editor, volume or number, month, note, organization, pages, publisher, series, type
manual	title	author, address, edition, month, note, organization, year
mastersthesis	author, title, school, year	address, month, note, type
misc	—	author, howpublished, month, note, title, year
phdthesis	author, title, school, year	address, month, note, type
proceedings	title, year	editor, organization, address, volume or number, series, month, publisher, note
techreport	author, title, institution, year	type, number, address, month, note
unpublished	author, title, note	month, year

```
@book{goossens97,
   author = "Goossens, Michel and Rahtz, Sebastian and
             Mittelbach, Frank",
   title = "The \LaTeX\ graphics companion: illustrating
            documents with \TeX\ and {PostScript}",
   publisher = "Addison Wesley Longman, Inc",
   year = 1997
}
```

In this example, the *keyword* is `goossens97`, so you cite the entry using the command `\cite{goossens97}`. The bibliography style usually converts the title to lowercase, so the name PostScript is enclosed in curly braces to prevent this happening.

Note that curly braces `{}` can be used instead of double quotes. The above example can just as easily be written:

```
@book{goossens97,
   author = {Goossens, Michel and Rahtz, Sebastian and
             Mittelbach, Frank},
   title = {The \LaTeX\ graphics companion: illustrating
            documents with \TeX\ and {PostScript}},
   publisher = {Addison Wesley Longman, Inc},
   year = 1997
}
```

Numbers (such as the year 1997) don't need to be delimited with quotes or braces. So you can have

```
pages = 10
```

but a page range would need to be written:

```
pages = "10--45"
```

Bibliography styles always have three-letter abbreviations for months: `jan`, `feb`, `mar`, etc. These should be used instead of typing them in explicitly, as their format depends on the bibliography style. These abbreviations should be entered without quotes. e.g.:

```
@inproceedings{talbot97,
   author    = "Talbot, Nicola and Cawley, Gavin",
   title     = "A fast index assignment algorithm for
                robust vector quantisation of image data",
   booktitle = "Proceedings of the I.E.E.E. International
                Conference on Image Processing",
   address   = "Santa Barbara, California, USA",
   month     = oct,
   year      = 1997
}
```

The following is an example of a bibliography database (you can download it if you want):

```
@book{goossens97,
   author = "Goossens, Michel and Rahtz, Sebastian and
             Mittelbach, Frank",
   title = "The \LaTeX\ graphics companion: illustrating
            documents with \TeX\ and {PostScript}",
   publisher = "Addison Wesley Longman, Inc",
   year = 1997
}

@inproceedings{talbot97,
   author    = "Talbot, Nicola L. C. and Cawley, Gavin C.",
   title     = "A fast index assignment algorithm for
                robust vector quantisation of image data",
```

```
   booktitle = "Proceedings of the I.E.E.E. International
                Conference on Image Processing",
   address   = "Santa Barbara, California, USA",
   month     = oct,
   year      = 1997
}

@article{cawley96,
   author    = "Cawley, Gavin C. and Talbot, Nicola L. C.",
   title     = "A fast index assignment algorithm for vector
                quantization over noisy transmission channels",
   journal   = "I.E.E. Electronic Letters",
   number    = 15,
   volume    = 32,
   pages     = "1343--1344",
   month     = jul,
   year      = 1996
}

@incollection{wainwright93,
   author    = "Wainwright, Robert B.",
   title     = "Hazards from {Northern} Native Foods",
   booktitle = "\emph{Clostridium botulinum}: Ecology and
                Control in Foods",
   chapter   = 12,
   pages     = "305--322",
   editor    = "Hauschild, Andreas H. W. and Dodds,
                Karen L.",
   publisher = "Marcel Dekker, Inc",
   year      = 1993
}
```

Once you have set up your bibliography database, you will need to first LaTeX your document, then call BibTeX and then LaTeX your document twice to get all the cross references up to date. If you are using TeXnicCenter, when you create a new project, click on the 'Uses BiBTeX' option, and it will automatically call BibTeX when you click on the build icon. If you are using a command prompt, then if your file is called, say, `thesis.tex`, you will need to type the following commands:

```
latex thesis
bibtex thesis
latex thesis
latex thesis
```

Note that you are specifying the auxiliary file when calling BibTeX. You can have a bibliography database that has a different name from your LaTeX file, but you use the name of the LaTeX file when calling BibTeX. For example, if your thesis is saved in the file `thesis.tex` and your bibliography database is saved in the file `ref.bib`, then you still need to do:

```
latex thesis
bibtex thesis
latex thesis
latex thesis
```

In fact, you can use multiple bibliography databases. Suppose your references are defined in the files `ref1.bib` and `ref2.bib`, then you need two `\bibliography` commands in `thesis.tex`:

```
\bibliography{ref1}
\bibliography{ref2}
```

Illustrations of the different bibliography styles are shown in Figures 5.1, 5.2, 5.3, 5.4, 5.5, 5.6 and 5.7. Note that the `apalike` bibliography style requires the `apalike` package.

Bibliography

[1] G. C. Cawley and N. L. C. Talbot. A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters*, 32(15):1343–1344, July 1996.

[2] M. Goossens, S. Rahtz, and F. Mittelbach. *The LaTeX graphics companion: illustrating documents with TeX and PostScript.* Addison Wesley Longman, Inc, 1997.

[3] N. L. C. Talbot and G. C. Cawley. A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA, Oct. 1997.

[4] R. B. Wainwright. Hazards from Northern native foods. In A. H. W. Hauschild and K. L. Dodds, editors, Clostridium botulinum*: Ecology and Control in Foods*, chapter 12, pages 305–322. Marcel Dekker, Inc, 1993.

Figure 5.1: `abbrv` bibliography style

Bibliography

[1] Cawley, G. C., and Talbot, N. L. C. A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters 32*, 15 (July 1996), 1343–1344.

[2] Goossens, M., Rahtz, S., and Mittelbach, F. *The LaTeX graphics companion: illustrating documents with TeX and PostScript.* Addison Wesley Longman, Inc, 1997.

[3] Talbot, N. L. C., and Cawley, G. C. A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing* (Santa Barbara, California, USA, Oct. 1997).

[4] Wainwright, R. B. Hazards from Northern native foods. In Clostridium botulinum*: Ecology and Control in Foods*, A. H. W. Hauschild and K. L. Dodds, Eds. Marcel Dekker, Inc, 1993, ch. 12, pp. 305–322.

Figure 5.2: `acm` bibliography style

Bibliography

[CT96] Gavin C. Cawley and Nicola L. C. Talbot. A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters*, 32(15):1343–1344, July 1996.

[GRM97] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX graphics companion: illustrating documents with TeX and PostScript*. Addison Wesley Longman, Inc, 1997.

[TC97] Nicola L. C. Talbot and Gavin C. Cawley. A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA, October 1997.

[Wai93] Robert B. Wainwright. Hazards from Northern native foods. In Andreas H. W. Hauschild and Karen L. Dodds, editors, Clostridium botulinum*: Ecology and Control in Foods*, chapter 12, pages 305–322. Marcel Dekker, Inc, 1993.

Figure 5.3: `alpha` bibliography style

Bibliography

[1] M. Goossens, S. Rahtz, and F. Mittelbach, *The LaTeX graphics companion: illustrating documents with TeX and PostScript*. Addison Wesley Longman, Inc, 1997.

[2] N. L. C. Talbot and G. C. Cawley, "A fast index assignment algorithm for robust vector quantisation of image data," in *Proceedings of the I.E.E.E. International Conference on Image Processing*, (Santa Barbara, California, USA), Oct. 1997.

[3] G. C. Cawley and N. L. C. Talbot, "A fast index assignment algorithm for vector quantization over noisy transmission channels," *I.E.E. Electronic Letters*, vol. 32, pp. 1343–1344, July 1996.

[4] R. B. Wainwright, "Hazards from Northern native foods," in Clostridium botulinum*: Ecology and Control in Foods* (A. H. W. Hauschild and K. L. Dodds, eds.), ch. 12, pp. 305–322, Marcel Dekker, Inc, 1993.

Figure 5.4: `ieeetr` bibliography style

Bibliography

[1] Gavin C. Cawley and Nicola L. C. Talbot. A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters*, 32(15):1343–1344, July 1996.

[2] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX graphics companion: illustrating documents with TeX and PostScript.* Addison Wesley Longman, Inc, 1997.

[3] Nicola L. C. Talbot and Gavin C. Cawley. A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA, October 1997.

[4] Robert B. Wainwright. Hazards from Northern native foods. In Andreas H. W. Hauschild and Karen L. Dodds, editors, Clostridium botulinum*: Ecology and Control in Foods*, chapter 12, pages 305–322. Marcel Dekker, Inc, 1993.

Figure 5.5: `plain` bibliography style

Bibliography

[1] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX graphics companion: illustrating documents with TeX and PostScript.* Addison Wesley Longman, Inc, 1997.

[2] Nicola L. C. Talbot and Gavin C. Cawley. A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA, October 1997.

[3] Gavin C. Cawley and Nicola L. C. Talbot. A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters*, 32(15):1343–1344, July 1996.

[4] Robert B. Wainwright. Hazards from Northern native foods. In Andreas H. W. Hauschild and Karen L. Dodds, editors, Clostridium botulinum*: Ecology and Control in Foods*, chapter 12, pages 305–322. Marcel Dekker, Inc, 1993.

Figure 5.6: `unsrt` bibliography style

Bibliography

Cawley, G. C. and Talbot, N. L. C. (1996). A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters*, 32(15):1343–1344.

Goossens, M., Rahtz, S., and Mittelbach, F. (1997). *The LaTeX graphics companion: illustrating documents with TeX and PostScript*. Addison Wesley Longman, Inc.

Talbot, N. L. C. and Cawley, G. C. (1997). A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA.

Wainwright, R. B. (1993). Hazards from Northern native foods. In Hauschild, A. H. W. and Dodds, K. L., editors, Clostridium botulinum*: Ecology and Control in Foods*, chapter 12, pages 305–322. Marcel Dekker, Inc.

Figure 5.7: `apalike` bibliography style (requires `apalike` package)

5.1 Back-References

The `backref` package supplied with the `hyperref` bundle will place a comma-separated list of section or page numbers on which the work was cited at the end of each item in the bibliography. Each bibliography item in the `thebibliography` environment must be separated by a blank line, but as BibTeX does this automatically, you only need to worry about it if you are creating your `thebibliography` environment without the aid of BibTeX.

The list of numbers will by default refer to the section numbers in which the corresponding `\cite` commands are located, but this can be changed to the page numbers by passing the option `pagebackref` to the `backref` package (or the `hyperref` package if you are using it.)

The `backrefx` package extends the `backref` package and provides additional text, such as: (Cited on pages 1, 4 and 10). Commands are available to modify the text produced. The style of output is illustrated in this document's bibliography.

5.2 Troubleshooting

- BibTeX writes `thebibliography` environment to a `.bbl` file. If you have made a LaTeX error in the `.bib` file, this error will be copied to the `.bbl` file. If you have corrected the error in the `.bib` file, but you are still getting an error when you LaTeX your document, try deleting the `.bbl` file.
- Remember to use double quotes or braces to delimit the field names in your `.bib` file.
- Remember to put a comma at the end of each field (except the last).
- Make sure you only have alphanumerical characters in the keywords.
- The LaTeX comment symbol (`%`) is not a comment character in a `.bib` file.
- If you have entered a field in the `.bib` file, but it doesn't appear in the bibliography, check to make sure that the field is required or optional for that type of entry, and check the spelling.

Chapter 6

Formatting

6.1 Double Spacing

Double spacing is usually frowned upon in the world of modern typesetting, however it is usually a requirement when you are writing a PhD thesis as it gives the examiners extra space to write comments.

Double spacing can be achieved either by using the `spacing` environment defined in the `doublespace` package, or by redefining the value of `\baselinestretch`. The value depends on the font size (see Table 6.1). To switch back to single spacing set `\baselinestretch` back to 1.

Table 6.1: Double Spacing Values for `\baselinestretch`

Font Size	10pt	11pt	12pt
`\baselinestretch`	1.67	1.62	1.66

For example, if you are using 12pt font, you will need the following line:

```
\renewcommand{\baselinestretch}{1.66}
```

6.2 Changing the Title Page

The title page style generated by `\maketitle` may not be appropriate for the school/university's specifications. If this is the case, you can use the `titlepage` environment instead. For example:

```
\begin{titlepage}
\begin{center}
\vspace*{1in}
{\LARGE A Sample PhD Thesis}
\par
\vspace{1.5in}
{\large A. N. Other}
\par
\vfill
A Thesis submitted for the degree of Doctor of Philosophy
\par
\vspace{0.5in}
School of Computing Sciences
\par
\vspace{0.5in}
University of East Anglia
\par
\vspace{0.5in}
July 2004
\end{center}
\end{titlepage}
```

The resulting output is shown in Figure 6.1.

Check with your supervisor to see if there is a particular layout required for the title page.

6.3 Verbatim Text

There may be times when you want to include text exactly as you have typed it into your source code. For example, you may want to include a short segment of computer code. This can be done using the `verbatim` environment. For example:

```
\begin{verbatim}
#include <stdio.h>

int main()
{
   printf{"Hello World\n"};

   return 1;
}
\end{verbatim}
```

would produce the following output:

```
#include <stdio.h>

int main()
{
   printf{"Hello World\n"};

   return 1;
}
```

The contents of a file can be included verbatim using the command

\verbatiminput{*filename*}

Definition

defined in the `verbatim` package. For example:

```
\verbatiminput{helloW.c}
```

where `helloW.c` is the filename (remember to use a forward slash `/` as a directory divider, even if you are using Windows).

Note: it is not usually appropriate to have a lot of listings in your thesis. It can annoy an examiner if you have included every single piece of code you have written during your PhD, as it comes across as padding to make it look as though your thesis is a lot larger than it really is. (Examiners are not easily fooled, and it's best not to irritate them as it is likely to make them less sympathetic towards you.) If you want to include listings in your thesis, check with your supervisor first to find out whether or not it is appropriate.

6.4 Tabbing

The `tabbing` environment lets you create tab stops so that you can tab to a particular distance from the left margin. Within the tabbing environment, you can use the command `\=` to set a tab stop, `\>` to jump to the next tab stop, `\<` to go back a tab stop, `\+` shifts the left border by one tab stop to the right, `\-` shifts the left border by one tab stop to the left, `\\` will start a new line and `\kill` will set any tabs stops defined in that line, but will not typeset the line itself.

Examples:

A Sample PhD Thesis

A. N. Other

A Thesis submitted for the degree of Doctor of Philosophy

School of Computing Sciences

University of East Anglia

July 2004

Figure 6.1: Example Title Page

1. This first example sets up three tab stops:

```
\begin{tabbing}
Zero \=One \=Two \=Three\\
\>First tab stop\\
\>A\>\>B\\
\>\>Second tab stop
\end{tabbing}
```

This produces the following output:

Zero One Two Three
First tab stop
A B
Second tab stop

2. This second example sets up four tab stops, but ignores the first line:

```
\begin{tabbing}
AAA \=BBBB \=XX \=YYYYYY \=Z \kill
\>\>\>Third tab stop\\
\>a \>\>b \>c
\end{tabbing}
```

This produces the following output:

Third tab stop
a b c

6.5 Theorems and Algorithms

A PhD thesis often contain theorems, lemmas, definitions etc. These structures can be created using the command

\newtheorem{*type*}{*title*}[outer counter]

Definition

where *type* is the type of your structure (e.g. theorem), *title* is the word that is printed in bold at the start of the structure (e.g. Theorem) and if the optional argument *outer counter* is present, then the structure's counter should depend on *outer counter* (as in the optional argument to \newcounter).

You should typically define your new theorem either in the preamble or in a package or class file. Once you have defined your new theorem, a new environment is created whose name is given by *type*. This environment has an optional argument that you can use to specify a caption for the structure.

Examples:

1. Define a theorem structure. The counter belonging to this structure is not dependent on any other counter:

```
\newtheorem{theorem}{Theorem}

\begin{theorem}
If $\lambda$ is an eigenvalue of $\mathbf{B}$ with
eigenvector $\vec{\xi}$, then $\lambda^n$ is an
eigenvalue of $\mathbf{B}^n$ with eigenvector $\vec{\xi}$.
\end{theorem}
```

This gives the following output:

Theorem 1 *If λ is an eigenvalue of $\mathbf{B}$ with eigenvector $\boldsymbol{\xi}$, then λ^n is an eigenvalue of $\mathbf{B}^n$ with eigenvector $\boldsymbol{\xi}$.*

(See LaTeX for Complete Novices [5] if you don't know how to redefine the \vec command so that the vector appears in bold.)

2. In this example, the theorem is defined to be dependent on the chapter counter. The theorem counter will be reset each time a new chapter is started:

```
\newtheorem{theorem}{Theorem}[chapter]

\begin{theorem}
If $\lambda$ is an eigenvalue of $\mathbf{B}$ with
eigenvector $\vec{\xi}$, then $\lambda^n$ is an
eigenvalue of $\mathbf{B}^n$ with eigenvector $\vec{\xi}$.
\end{theorem}
```

This gives the following output:

Theorem 6.1 *If λ is an eigenvalue of $\mathbf{B}$ with eigenvector $\boldsymbol{\xi}$, then λ^n is an eigenvalue of $\mathbf{B}^n$ with eigenvector $\boldsymbol{\xi}$.*

3. In this example, the theorem is given a caption:

```
\newtheorem{theorem}{Theorem}[chapter]

\begin{theorem}[Eigenvector Powers]
If $\lambda$ is an eigenvalue of $\mathbf{B}$ with
eigenvector $\vec{\xi}$, then $\lambda^n$ is an
eigenvalue of $\mathbf{B}^n$ with eigenvector $\vec{\xi}$.
\end{theorem}
```

This gives the following output:

Theorem 6.1 (Eigenvector Powers) *If λ is an eigenvalue of $\mathbf{B}$ with eigenvector $\boldsymbol{\xi}$, then λ^n is an eigenvalue of $\mathbf{B}^n$ with eigenvector $\boldsymbol{\xi}$.*

4. In this example an algorithm structure is created. The commands `\hfill\par` are used to prevent the `tabbing` environment from running into the algorithm title.

```
\newtheorem{algorithm}{Algorithm}

\begin{algorithm}[Gauss-Seidel Algorithm]
\hfill\par
\begin{tabbing}
1. \=For $k=1$ to maximum number of iterations\\
\>2. For \=$i=1$ to $n$\\
\>\>Set
\begin{math}
x_i^{(k)} =
\frac{b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}
         -\sum_{j=i+1}^{n}a_{ij}x_j^{(k-1)}}%
     {a_{ii}}
\end{math}
\\
\>3. If $\|\vec{x}^{(k)}-\vec{x}^{(k-1)}\| < \epsilon$,
where $\epsilon$ is a specified stopping criteria, stop.
\end{tabbing}
\end{algorithm}
```

This will give the following output:

Algorithm 1 (Gauss-Seidel Algorithm)

1. *For* $k = 1$ *to maximum number of iterations*
   2. *For* $i = 1$ *to* n
      Set $x_i^{(k)} = \frac{b_i-\sum_{j=1}^{i-1} a_{ij}x_j^{(k)}-\sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}}{a_{ii}}$
   3. *If* $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\| < \epsilon$, *where* ϵ *is a specified stopping criteria, stop.*

The last example doesn't look right, as algorithms tend to be displayed in an upright font not an italic font. The package `amsthm` extends the functionality of `\newtheorem` and provides three theorem styles:

plain Title and number in bold, body in italic (default).

definition Title and number in bold, body in normal font.

remark Title and number in italic, body in normal font.

The above example can now be changed to:

```
\theoremstyle{definition}
\newtheorem{algorithm}{Algorithm}

\begin{algorithm}[Gauss-Seidel Algorithm]
\hfill\par
\begin{tabbing}
1. \=For $k=1$ to maximum number of iterations\\
\>2. For \=$i=1$ to $n$\\
\>\>Set
\begin{math}
x_i^{(k)} =
\frac{b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}
         -\sum_{j=i+1}^{n}a_{ij}x_j^{(k-1)}}%
     {a_{ii}}
\end{math}
\\
\>3. If $\|\vec{x}^{(k)}-\vec{x}^{(k-1)}\| < \epsilon$,
where $\epsilon$ is a specified stopping criteria, stop.
\end{tabbing}
\end{algorithm}
```

This will give the following output:

Algorithm 1 (Gauss-Seidel Algorithm)

1. For $k = 1$ to maximum number of iterations
   2. For $i = 1$ to n
      Set $x_i^{(k)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}}{a_{ii}}$
   3. If $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\| < \epsilon$, where ϵ is a specified stopping criteria, stop.

(You can download an example of this.)

Alternatively, if you want your algorithms to behave like figures and tables, you can use the `\newfloat` command defined in the `float` package (by Anselm Lingnau):

`\newfloat{`*type*`}{`*placement*`}{`*ext*`}[`*outer counter*`]`

Definition

where *type* is the name of your new float, *placement* is the default placement specifier (`t`, `b`, `p` and `h`), *ext* is the extension for the list of *type* and as before, the presence of *outer counter* indicates that the counter associated with this new float should depend on *outer counter*.

You can also specify the style of your new floats by issuing the command:

`\floatstyle{`*style*`}`

Definition

before defining your new floats, where *style* can be one of:

plain Same as the standard `figure` and `table` floats, except that the caption is always placed at the end of the float.

boxed The body of the float is placed in a box, and the caption is printed below the box.

ruled The caption is printed at the top with a rule above and below it, and there is a rule at the end of the float.

The name associated with a float is defined using the command:

`\floatname{`*type*`}{`*name*`}`

Definition

where *type* is the name of the float environment (as defined in \newfloat) and *name* is the name associated with that float.

The list of *type* can be produced using the command:

\listof{*type*}{*title*}

Definition

So, instead of defining our algorithm environment using \newtheorem, we could instead define it using \newfloat as follows:

```
\floatstyle{ruled}
\newfloat{algorithm}{htbp}{loa}
\floatname{algorithm}{Algorithm}

\begin{algorithm}
\caption{Gauss-Seidel Algorithm}
\label{alg:GS}

\begin{tabbing}
1. \=For $k=1$ to maximum number of iterations\\
\>2. For \=$i=1$ to $n$\\
\>\>Set
\begin{math}
x_i^{(k)} =
\frac{b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}
         -\sum_{j=i+1}^{n}a_{ij}x_j^{(k-1)}}{a_{ii}}
\end{math}
\\
\>3. If $\|\vec{x}^{(k)}-\vec{x}^{(k-1)}\| < \epsilon$,
where $\epsilon$ is a specified stopping criteria, stop.
\end{tabbing}
\end{algorithm}
```

This would produce the following output:

Algorithm 1 Gauss-Seidel Algorithm

1. For $k = 1$ to maximum number of iterations
   2. For $i = 1$ to n
      Set $x_i^{(k)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}}{a_{ii}}$
   3. If $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\| < \epsilon$, where ϵ is a specified stopping criteria, stop.

The following line can then go after the list of figures and list of tables:

```
\listof{algorithm}{List of Algorithms}
```

(You can download an example of this.)

Chapter 7

Generating an Index or a Glossary

It is fairly straight-forward to create an index or glossary using LaTeX, and using the `makeindex` application makes it even easier. It is a good idea to include a glossary in a thesis, particularly if there is any mathematics in your work, as there are a number of different ways some symbols can be interpreted. For example, x' could mean $\frac{dx}{dt}$ or it could mean an updated value of x (or it could even mean the transpose of x, but in this case x should be formatted as a vector.) It is not wise to assume that your reader uses the same notation as you. It isn't quite so common to include an index in a PhD thesis, however, the LaTeX user's guide [3] states that any nonfiction work of more than twenty pages ought to have an index. If you are only interested in creating a glossary, I would recommend that you still read how to generate an index as they have a similar form.

7.1 Generating an Index

If you want to generate an index, you will need the command `\makeindex` in the preamble. The command

`\index{`*entry*`}`

Definition

is used to index *entry* at that point in the document. For example, the following code:

```
Eigenvectors\index{eigenvector} are defined \ldots
```

will produce the output

Eigenvectors are defined ...

and place the entry 'eigenvector' in the `.idx` file with the associated page number.

The package `makeidx` provides the command `\printindex` which should be placed at the point in the document where you want your index to appear. The command `\makeindex` will cause each `\index` command to write the relevant information to the `.idx` file. This file can then be processed by `makeindex` to produce a `.ind` file which contains a `theindex` environment. This file is then read in by the `\printindex` command on the next LaTeX run. If you are using `TeXnicCenter` you will need to select "uses makeindex" when you create a new project, if you are using a command prompt, you will need to do:

```
latex filename.tex
makeindex filename.idx
latex filename.tex
```

(where `filename` is the name of your file, e.g. `thesis`) If you are also using BibTeX, you will need to do:

```
latex filename.tex
bibtex filename
makeindex filename.idx
```

```
latex filename.tex
latex filename.tex
```

It's a good idea to have sub-entries within an index, to assist the reader. For example, you may want to index the term "matrix", but your document may mention many different types of matrices, such as diagonal, block or singular. In which case it would be better to index the term matrix for general occurrences, and have sub-entries indexing specific types of matrices, so that the matrix entry in the index would look something like:

matrix, 4, 10, 22–24

diagonal, 12

block, 20, 24

singular, 33

A sub-entry can be generated using the `!` character. So the above can be generated using the following commands:

Preamble:	`\makeindex`
Page 4:	`\index{matrix}`
Page 10:	`\index{matrix}`
Page 12:	`\index{matrix!diagonal}`
Page 20:	`\index{matrix!block}`
Page 22:	`\index{matrix}`
Page 23:	`\index{matrix}`
Page 24:	`\index{matrix}`
Page 24:	`\index{matrix!block}`
Page 33:	`\index{matrix!singular}`
End of Doc:	`\printindex`

Note that the same entries on pages 22, 23 and 24 have been turned into a page range 22–24. For larger page ranges, you can specify the start of the page range by appending `|(` to the end of the index entry and the end of the page range by appending `|)` to the end of the index entry. For example:

Preamble:	`\makeindex`
Page 4:	`\index{matrix}`
Page 10:	`\index{matrix}`
Page 12:	`\index{matrix!diagonal}`
Page 20:	`\index{matrix!block}`
Page 22:	`\index{matrix|(}`
Page 24:	`\index{matrix!block}`
Page 30:	`\index{matrix|)}`
Page 33:	`\index{matrix!singular}`
End of Doc:	`\printindex`

would produce the following output in the index:

matrix, 4, 10, 22–30

diagonal, 12

block, 20, 24

singular, 33

An index entry can refer to another entry using `|see{`*reference*`}`. For example,

```
\index{singular matrix|see{matrix, singular}}
```

would produce the entry

singular matrix, *see* matrix, singular

The format of the page number can be changed using `|`*style* where *style* is the name of a formatting command *without* the backslash. Suppose in the above example, the definition of a matrix is specified on page 10, then you may want the page number to appear in bold to indicate that this is a primary reference. The command `\textbf` produces bold text, so you would need to append `|textbf` to the index entry. For example, the code:

Preamble:	`\makeindex`
Page 4:	`\index{matrix}`
Page 10:	`\index{matrix|textbf}`
Page 12:	`\index{matrix!diagonal}`
Page 20:	`\index{matrix!block}`
Page 22:	`\index{matrix|(}`
Page 24:	`\index{matrix!block}`
Page 30:	`\index{matrix|)}`
Page 33:	`\index{matrix!singular}`
End of Doc:	`\printindex`

would produce the following output in the index:

matrix, 4, **10**, 22–30

diagonal, 12

block, 20, 24

singular, 33

The application `makeindex` sorts the index according to the entries specified, so the word "matrix" would come before the word "modulus", but `$\mu$` will be sorted on the characters `$`, `\`, `m`, `u` and then `$`, so μ would come before "matrix". This may not be appropriate, so it is possible to specify how to sort the entry and how to format the entry separately using the `@` character:

```
\index{mu@$\mu$}
```

In this case the sorting is performed on the string `mu`, so it will appear after the word "modulus", but it will appear in the index as μ. For more information about generating an index see the LaTeX user's guide [3], *The LaTeX Companion* [1] or *A Guide to LaTeX* [2].

7.1.1 Troubleshooting

- My index hasn't appeared.

  1. Make sure you have the command `\printindex` at the place where you want the index to appear (this command is defined in the `makeidx` package).
  2. Make sure you have the command `\makeindex` in the preamble.
  3. Make sure you LaTeX the document, then run `makeindex`, then LaTeX the document again.

- I want to index the character `@`, `!` or `|` but it's not working.

  If you want any of these symbols in your index, you will need to prepend the character with the double quote symbol `"`. For example:

  ```
  \index{"@}
  ```

  will index the `@` character.

- I have multiple entries of the same item. For example:

  matrix, 10, 22-30

  matrix, 4

  Check to make sure the sort argument to each of the corresponding `\index` commands is the same, pay particular attention to spaces as `makeindex` will treat the following entries differently:

  ```
  \index{matrix}
  \index{ matrix}
  \index{matrix }
  ```

7.2 Generating a Glossary

There are a number of packages available to assist creating a glossary, these include `makeglos` (analogous to `makeidx`), `glossary`, `glosstex` and `gloss`. The first two use LaTeX in conjunction with `makeindex`, the third (`glosstex`) uses LaTeX in conjunction with `makeindex` and `glosstex` whilst the fourth (`gloss`) uses LaTeX in conjunction with BibTeX. This document only describes `makeglos` and `glossary`, as they are similar in form to `makeidx`. If you are interested in using the others, you should read their accompanying documentation.

A glossary is produced in much the same way as an index, except that you use the command `\makeglossary` instead of `\makeindex` and the command `\glossary` instead of the command `\index`. Both `makeglos` and `glossary` provide the command `\printglossary`, analogous to `\printindex`

7.2.1 The `makeglos` Package

Consider the following example:

```
Preamble :  \makeglossary
Page 2 :    \glossary{set: a collection of objects}
Page 3 :    \glossary{cardinality: the number of objects within a set}
Page 4 :    \glossary{universal set: the set containing everything}
```

LaTeXing this document will create a file with the extension `.glo` which contains all the glossary details. You can use the `makeindex` application to process these entries, however you will need to make some modifications.

1. You will need to create a new `makeindex` style file that tells `makeindex` to search for `\glossaryentry` instead of `\indexentry` and to create a `theglossary` environment instead of a `theindex` environment. Let's call our new `makeindex` style file `thesisglo.ist`. We first need to set the keyword to `"\\glossaryentry"`:

   ```
   keyword "\\glossaryentry"
   ```

   we now need to change the preamble to `"\\begin{theglossary}\n"` and the postamble to `"\n\n\\end{theglossary}\n"`:

   ```
   preamble "\\begin{theglossary}\n"
   postamble "\n\n\\end{theglossary}\n"
   ```

   We now need to tell `makeindex` to use this style file using the `-s` option, and you also need to specify the output file, which should have a `.gls` extension, using the `-o` option:

   ```
   makeindex -o thesis.gls -s thesisglo.ist thesis.glo
   ```

   (This is assuming that the main document is contained in the file `thesis.tex` and that you have LaTeXed it prior to calling `makeindex`.) Note that you are now using `thesis.glo` (created by `\glossary` commands) as the input file, not `thesis.idx` (created by `\index`) commands.

2. By default, `makeindex` will use the file with the extension `.ilg` as the log file, you may want to change this to avoid a conflict with the index log file. For example, you may want to call the glossary log file `thesis.glg`:

   ```
   makeindex -t thesis.glg -o thesis.gls -s thesisglo.ist thesis.glo
   ```

Here's a simple example using `makeglos`:

File `sample.tex`:

```
\documentclass[a4paper]{report}
\usepackage{makeglos}
\makeglossary
\begin{document}
```

```
\printglossary

\chapter{Introduction}
A set\glossary{set: A collection of objects} is usually
denoted in a calligraphic font, e.g.\ $\mathcal{S}$.  The
cardinality\glossary{cardinality: The number of objects in a set}
of $\mathcal{S}$ is denoted $|\mathcal{S}|$.
The universal set\glossary{universal set: The set of everything}
is generally denoted $\mathcal{U}$
\end{document}
```

The `makeindex` style file, `sample.ist`, should look something like:

```
keyword "\\glossaryentry"
preamble "\\begin{theglossary}\n"
postamble "\\end{theglossary}\n"
```

You then need to do

```
latex sample.tex
makeindex -t sample.glg -o sample.gls -s sample.ist sample.glo
latex sample.tex
```

The glossary title (Glossary, by default) can be changed by redefining the command `\glossaryname`. If you want any text to appear at the start of the glossary, you can redefine the command `\glossaryintro`. The format of the argument to the `\glossary` command is the same as with `\index`, so you can use `@` to specify how to sort the entry, `|` to specify how to format the associated page number and `!` so indicate sub-entries (although this usually isn't appropriate for a glossary). If you have any problems, check the index troubleshooting section on page 30.

You can download the files thesis9.tex and thesisglo.ist which illustrate this example.

7.2.2 The `glossary` Package

The `glossary` package also defines the command `\printglossary`, but it redefines the `\glossary` command, so you can separate out the entry name and its corresponding description, using a set of *key=value* pairs. The following keys are available:

name	The entry name
description	A description of the entry
sort	How to sort the entry. (Entry name used by default)
format	How to format the page number

The example in the previous section can be changed to:

```
Preamble :  \makeglossary
Page 2 :    \glossary{name=set,description=a collection of
                      objects}
Page 3 :    \glossary{name=cardinality,description=the number of
                      objects within a set}
Page 4 :    \glossary{name=universal set,description=the set
                      containing everything}
```

The `glossary` package creates a `makeindex .ist` style file customised for your document, so you don't need to worry about creating it. By default, the name of the `.ist` file will have the same root as your document, so if your document is called, say `sample.tex`, the file `sample.ist` will be created when you pass `sample.tex` to LaTeX. So, as before, you will need to do:

```
latex sample.tex
makeindex -t sample.glg -o sample.gls -s sample.ist sample.glo
latex sample.tex
```

Alternatively, you can use the Perl script `makeglos` supplied with version 2.0 of the `glossary` package:

```
latex sample.tex
makeglos sample.glo
latex sample.tex
```

The style of the glossary can be customised. As before, the glossary title (Glossary, by default) can be changed by redefining the command `\glossaryname`. The glossary style can be changed using the package options, which should be in a *key=value* list form:

style The style of `theglossary` environment. Values:

- **list** use description environment in the glossary
- **super** use supertabular environment in the glossary
- **long** use longtable environment in the glossary (Default)

header Glossary header. Values:

- **none** The glossary doesn't have a heading (Default)
- **plain** The glossary has a heading

border Glossary border. Values:

- **none** The glossary doesn't have a border (Default)
- **plain** Border around the main body of the glossary

cols Number of columns. Values:

- **2** The entry name and description are in two separate columns with the associated page numbers in the same column as the description. (Default)
- **3** The entry name, description and associated page numbers are in three separate columns.

number Associated number corresponding to each entry value[1]. Values:

- **page** Each entry has the corresponding page number(s) where the entry is defined. (Default)
- **section** Each entry has the corresponding section number(s) where the entry is defined.
- **none** The corresponding numbers are suppressed.

toc Boolean variable[2]

- **true** Add glossary to table of contents
- **false** Don't add glossary to table of contents (Default)

Note that if you specify this option, you will need to run LaTeX twice after generating the glossary.

hyper Boolean variable[3]

- **true** Make associated numbers a hypertext link
- **false** Don't make associated numbers a hypertext link

If the `hyperref` package has been loaded prior to loading the `glossary` package, `hyper=true` is set, otherwise the default is `hyper=false`.

The `border`, `header` and `cols` options should not be used in conjunction with `style=list`, as they only make sense with one of the tabular-style options. Example:

```
\usepackage[style=long,cols=3,border=plain]{glossary}
```

If you want to add any extra information to the start or end of the glossary, you can redefine the commands `\glossarypreamble` and `\glossarypostamble`. It is also possible to define additional glossary style objects, so that you can have more than one type of glossary in your document. For example, a glossary of terms and an index of mathematical functions or symbols. See the `glossary` package documentation for details. The latest version of the `glossary` package can be downloaded from http://theoval.cmp.uea.ac.uk/~nlct/latex/packages/index.html#glossary.

You can download an example.

[1] this option is only available in version 1.1 and above
[2] this option is only available in version 2.0 and above
[3] this option is only available in version 2.0 and above

Chapter 8

Too Many Unprocessed Floats

A common problem PhD student's encounter when writing a thesis is the "too many unprocessed floats" error. This is usually caused by having too many figures and tables in the results chapter and not enough surrounding text. If this happens, there are a number of things you can try doing:

1. Make sure you haven't been too restrictive in where you want your floats to go. If you use a placement specifier, give LaTeX as many options as possible. For example:

   ```
   \begin{figure}[htbp]
   ```

   which indicates that the figure can be placed "here", at the top of a page, at the bottom of the page or on a page solely consisting of floats.

2. Try increasing the amount of text in the chapter. Remember that you should never simply print all the figures and tables in a results chapter without discussing them to some extent.

3. If all else fails, try using the `\clearpage` command. This forces all unprocessed floats to be processed immediately, and start a new page. This may result in the page ending prematurely, if you wish to avoid this, you can use David Carlisle's `afterpage` package, and use the command:

   ```
   \afterpage{\clearpage}
   ```

For other problems, check the FAQ on the TeX Archive [4].

Bibliography

[1] "The LaTeX Companion", Michel Goossens, Frank Mittelbach and Alexander Samarin, Addison-Wesley (1994). (Cited on pages 11, 12 and 30.)

[2] "A Guide to LaTeX2ε: document preparation for beginners and advanced users", Helmut Kopka and Patrick W. Daly, Addison-Wesley (1995). (Cited on pages 8, 10, 11, 12 and 30.)

[3] "LaTeX : a document preparation system", Leslie Lamport, 2nd ed. Addison-Wesley (1994). (Cited on pages 28 and 30.)

[4] The TeX Archive. http://www.tex.ac.uk/ (Cited on page 34.)

[5] "LaTeX for Complete Novices", Nicola Talbot. http://theoval.cmp.uea.ac.uk/~nlct/latex/novices/ (2004). (Cited on pages b, 1 and 24.)

Index