Using $\LaTeX$ for University of Utah Theses

`UUTHESIS.STY Version 1.8, February 1994`

Stan Shebs *

September 24, 2004

1 Introduction

This document[1] tells how to format a University of Utah thesis or dissertation using the computer typesetting system LaTeX. Formatting standards are under the control of a document style called `uuthesis`.

This guide is made to be used in conjunction with **A Handbook for Theses and Dissertations**, which is published by the graduate school of the University of Utah. It is recommended that you obtain a recent copy of the above handbook during a personal visit to the thesis office. Make this first visit *before* starting on a thesis or dissertation. The number of contacts with the thesis office averages five: one initial visit, one after chapter 1 is finished, some question and answer sessions followed by a reading before the thesis defense and a final reading after all revisions.

The thesis office, like professional journals in science and engineering, has their own set of publication standards. If you want to publish your work, then you have to meet the standards. In the case of the thesis office, the standards and requirements are printed in the *Handbook*.

For mathematics theses, the editors request that you and your thesis advisor decide on a style guide. This does not mean that you have settled upon *A Manual for Authors of Mathematical Papers* published by the American Mathematical Society. The latter is *not a style guide*. You are expected to choose *The ACS Style Guide* or *The Chicago Manual of Style*. Hybrids adopted from standard journals may also be acceptable.

For computer science theses, the same question will be asked: will it be the Chicago Manual or the ACS Guide? More manuals of style are listed in the references of the *Handbook*.

The basic problem for everyone who writes a thesis or dissertation is the arrangement of topics and results into parts and chapters, sections and subsections, appendices, tables and figures. The *rule book* for this process is the graduate school's *Handbook*. The `uuthesis` style was created to ease the job of meeting the requirements. It is part of the solution to the problem, but not the entire solution, and it is certainly not the *Handbook*.

*Updated by Grant Gustafson, 1988-1994.

[1] Updated by Grant Gustafson, 1988-1994.

Some knowledge of TEX and LATEX is assumed. If you haven't used either before, then format a simpler and shorter document before tackling a thesis. One document that will be useful is the test file in item 4 below.

After a review session with LATEX features, begin a serious study of the sample thesis prepared by Jeff McGough, Grant Gustafson and Nelson Beebe, 1993. See item 5 below.

Beginners with LATEX should obtain the following documents and sources:

1. Leslie Lamport, *LATEX User's Guide and Reference Manual*, Addison-Wesley Publishing Company, Reading, Massachusetts (1986). ISBN 0-201-15790X (LATEX version 2.09 released 19 April 1986).

2. Grant B. Gustafson, *LATEX Article Style*, Department of Mathematics, University of Utah, SLC, UT 84112 (801–581–6851), 1988, 1992, 50 pages. Copy from `/u/ma/gustafson/tex/art.tex`, or request via email: `gustafson@math.utah.edu`

3. The University of Utah thesis style files:

   UUGUIDE.TEX — source for this guide.
   UUTHESIS.STY — Thesis style file version 1.8

   These are installed and maintained on computer systems in Computer Science and The Department of Mathematics. Previous versions use sources `uut11.sty` and `uut12.sty`. These last sources are not used now since they are included in `uuthesis.sty`.

4. Small sample thesis test file available on mathematics department unix machines as the single source

   `/usr/local/lib/tex/uuthesis/testspacing.tex`

   The source uses `uuthesis.sty`. Copy that source also, if it is not available at your computer site. This source is small but contains figures and tables, bibliography, table of contents, list of figures, list of tables, all the front matter, acknowledgements, abstract, and uses all the sectioning commands. It is useful as an initial test file when setting up your thesis and also as a debugging source when you have problems. Use it to understand figure placement and basic format of a thesis. This source can be obtained by email. Send your request to Beebe or Gustafson at the address listed at the end of this guide.

5. Sample thesis file available on mathematics department unix machines as sources in root file

   `/usr/local/lib/tex/uuthesis/samplethesis/thesis.tex`

   There are many parts to this sample thesis. Copy the entire directory and examine it closely to understand how a thesis is managed. If you cannot access this directory, then request email copies or copies on diskette. Send your request to Beebe or Gustafson at the address listed at the end of this guide.

6. Sample Computer Science and Mathematics theses, by Stan Shebs (CS), David Eyre (Math), Paul Joyce (Math) and Jeff McGough (Math). These documents are available through the University Library and serve as specific examples of how to organize a thesis. The sources to these may also be available; contact the people listed at the end of this guide.

7. Documentation for `xfig`, `fig2dev`, `gnuplot`, `maple`. These unix programs are invaluable for making figures and drawings in LaTeX using `latex` or `pictex` macros. Also learn about `graph` and `x79` for manipulation of plots, and `pltde` and `phase` for solving and plotting differential equations.

Some additional references are available, related to thesis production, although not in so direct a manner:

1. Nelson H. F. Beebe, *A Bibliography of Publications about TeX*, 1993. Available by anonymous FTP from site `science.utah.edu` in directory `pub/tex/bib`, files `texbook1.bib` and texbook1.ltx .
2. David J. Buerger, *LaTeX for Engineers and Scientists*, McGraw-Hill, New York (1990), ISBN 0-07-008845-4.[2]
3. Jane Hahn, *LaTeX for Everyone*, PTI, 1991.[3]
4. Michael D. Spivak, *The Joy of TeX — A Gourmet Guide to Typesetting with the AmsTeX macro package*, AMS, 2nd revised edition, 1990, ISBN 0-8218-2997-1.

2 The Document Style

The `uuthesis` document style is largely adapted from the `report` style with *doublespace* option. `Uuthesis` shares many of the characteristics of the `report` style. For instance, double-sided formatting is available (for convenience), but double columns are not. The 12-point font happens by default. An 11-point is available by specifying the `11pt` option, which can cut many pages off a long thesis. No other type sizes are available. Remarks hereafter are about the 12pt option. The main text of the thesis will be double-spaced automatically, which results in `baselineskip` of about 21pt (72.27pt per inch). If you need to insert single-spaced text, then use the `singlespace` environment, which results in a baselineskip of 13.5pt. Certain pre-defined options in the `uuthesis` format already use the `singlespace` environment and therefore only wizards should change the basic baselineskip dimensions.

In cases where the Thesis Manual is not explicit, the style follows the *Chicago Manual*.

2.1 Document Style Options

There are several options that may be supplied:

11pt Set the text in 11pt type. This size uses 5pt fonts for `tiny`, which is less than the required 2mm font height for University Microfilms. Consequently, do not use `tiny` with the 11pt option (this can happen in math mode).

12pt Set the text in 12pt type (this is the default).

[2]Some basic ideas about spacing control in this book make poor advice for a thesis. It is a useful reference that is perhaps easier to read than Lamport's *Manual*, since it is not as energetic and does not qualify at all as a reference manual.

[3]Recommended for beginners in LaTeX.

twoside Format the pages for two-sided printing.

draft Relax the requirements on box overflow. Do *not* use this for what the thesis editor will see!

report Format things appropriately for a technical report. At present, this just means defaulting to single spacing, omitting the signature pages, and using the command `reporttitlepage` to generate the title page (the builtin definition is only appropriate for Computer Science, but can be redefined with no problems).

3 What the Style Defines

This section lists all the control sequences defined or modified by the style. Later sections will describe these in more detail.

The following standard LaTeX environments are modified by this style:

`description` environment
`equation` environment
`figure` environment
`quotation` environment
`quote` environment
`verse` environment
`table` environment
`thebibliography` environment

Although these environments are generally similar to those in other LaTeX style, it is prudent to experiment with the thesis style before assuming anything about the output.

In addition, there are some declarations for new environments, some of which are in common use in mathematics papers. The details may be found in a later section.

`Proof` environment
`singlespace` environment
`doublespace` environment
`normalspace` environment
`index` environment
`epigraph` environment
`topics` environment

The following optional environments are enabled by the control string `\theoremsetup` in the preamble. They are undefined if the control string does not appear in the preamble.

`theorem` environment
`proposition` environment
`corollary` environment

Many new control sequences are defined, and a number of standard ones modified. None of those listed here should be modified unless you know what you are doing and are willing to experiment. Arguments to the control sequences below are indicated as `#1`, `#2`, and so forth. These arguments are usually ordinary text, but in a few special cases there are extreme limitations on content.

Modified standard LaTeX commands:

```
\appendix
\author{#1}
\part{#1}
\chapter{#1}
\section{#1}
\subsection{#1}
\subsubsection{#1}
\subsubsubsection{#1}
\paragraph{#1}
\index
\subitem
\subsubitem
\titlepage
\thebibliography{#1}
\listoffigures
\listoftables
\tableofcontents
```

(`\chapter{#1}` — **Warning**: Read about this one!)

New commands (the arguments are never optional):

```
\abstracttitlepage
\chairtitle{#1}
\captionONfalse
\captionONtrue
\captionlineskip
\committeeapproval
\committeechair{#1}
\copyrightpage
\copyrightyear{#1}
\dedicationpage
\dedication{#1}
\degree{#1}
\departmentchair{#1}
```

```
\department{#1}
\firstreader{#1}
\fivelevels
\fixchapterheading
\fourlevels
\fourthreader{#1}
\frontmatter{#1}{#2}{#3}
\frontmatterformat
\graduatedean{#1}
\listoffiguresfalse
\listoffigures
\listoftablesfalse
\listoftables
\mainheader{#1}
\mainheadingwidth
\maintext
\minusfourthline
\minushalfline
\minusline
\noisyfalse
\noisytrue
\normalspace
\optionalfront{#1}{#2}
\plusfourthline
\plushalfline
\plusline
\prefacesection{#1}
\preface{#1}{#2}
\pf
\proof{#1}
\proofline{#1}
\qed
\rawbibliographytrue
\rawbibliographyfalse
\readingapproval
\reportitlepage
\secondreader{#1}
\singlespace
\submitdate{#1}
\tableofcontents
\thesistype{#1}
\thirdreader{#1}
\threelevels
\titlepage
```

```
\twopagefigure{#1}{#2}
\ulabel{#1}{#2}
\vita
```

4 Front Matter

The Front Matter consists of the extra pages at the front of the thesis, such as the title page and reading approval pages. In particular, the chapters of the thesis are specifically excluded from the front matter. In the initial stages of writing, the front matter is unimportant, and you can just start with the `\maintext` command described in the next section.

Almost all of the front matter pages can be generated automatically, including the signature pages. A number of declarations must be supplied in the preamble, so that the appropriate places get filled in with the right information (all declarations have one argument #1, which is ordinary text). Declarations must appear in the preamble, that is, *after* `\documentstyle{uuthesis}` and *before* the control `\begin{document}`.

`\thesistype{type}` is the type of thesis. *Type* is either `thesis` for Master's theses, or `dissertation` for doctorates. The word goes through directly, so be sure of the spelling!

`\title{text}` declares the full title of your thesis. If it is too long to fit on one line, then designate line breaks (using \\) at appropriate locations.

`\author{names}` declares your full legal name (as the Thesis Manual says). Multiple authors will have to use explicit line breaks.

`\degree{name}` is the name of the degree. Sometimes the standard phrases are insufficient, such as when getting a specific degree from an omnibus department.

`\department{name}` is the full name of the department, for example, *Department of Mathematics*.

`\departmentchair{name}` is the name of the department chairperson.

`\submitdate{month year}` is the date of submission, which must be the end of a quarter, as specified in the manual.

`\copyrightyear{year}` is the year that should appear in the copyright notice.

`\committeechair{name}` is the name of your advisor or committee chairperson. Multiple names are unexpected and may not be processed correctly.

`\chairtitle{name}` is the title of your advisor or committee chairperson, usually *professor*.

`\firstreader{name}`

`\secondreader{name}`

`\thirdreader{name}`

`\fourthreader{name}` declares the names of committee members. The committee approval page will end up with the appropriate number of lines for signatures. The command `fifthreader` emits an error message and defaults back to four readers.

`\graduatedean{name}` is the name of the graduate dean (B. Gale Dick, as of this writing).

`\dedication{words}` declares a dedication. No special formatting, but you can use `\\` to get multiple lines at least. A multi-paragraph dedication is unlikely to work. The dedication page will be omitted if this declaration is not supplied.

You have two ways to format the front matter; the easy way and the hard way. The easy way is automatic, somewhat restrictive, but sufficient for 99% of all theses. The hard way gives more control, but takes more work to get right.

4.1 Automatic Front Matter

The "one command to do it all" is

```
\frontmatter{abstract}{acknowledge}{acktitle}
```

where `abstract` is the name of the LaTeX source file containing the abstract's text, `acknowledge` is the name of the LaTeX source file for the preface, and `acktitle` is the heading that will be used for the preface ("Preface" and "Acknowledgments" are the two most common choices). The `\frontmatter` command will generate all pages up to the beginning of the text proper. The abstract may not contain citations. Further, the University of Utah requires the total length of the abstract to be less than 350 words.

The lists of figures and tables will appear by default. To disable them, use the commands below:

```
\listoffiguresfalse
\listoftablesfalse
```

These commands eliminate the lists and any possible reference to the lists that might have appeared in the table of contents.

Lists are followed by the acknowledgments or preface, which are at the end of the front matter. The title is one of the arguments to the `\frontmatter` command. The title will always be capitalized.

4.2 Manual Front Matter

This method requires that you specify each front matter page individually. It is the most versatile. For example, it allows your thesis to include things like a glossary, if it needs one. The declarations are to be typed in the order below, just *after* `\begin{document}`:

`\frontmatterformat` This is required in order to set up spacing and page numbering for the front matter. Note that the page numbers are Roman in the front matter and arabic in the main thesis. If you get a strangely numbered table of contents, then suspect this control sequence was somehow omitted.

`\titlepage` Uses text set up by preamble commands *title, author, thesistype, department, degree.* Creates a separate title page with empty page style.

`\copyrightpage` Uses preamble commands *author, copyrightyear.* Creates a special copyright page required by the Thesis Office.

`\committeeapproval` Uses text set up by preamble commands. The commands are *thesistype*, *author*, *committeechair*, *firstreader*, *secondreader*, *thirdreader*, *fourthreader*. Produces a **Supervisory Committee Approval** form including signature lines. This form certifies that each member of the committee has read the thesis and that the thesis has been approved by majority vote. Signatures on this form are absolute: refusal to sign at the thesis defense is not unknown, because the form can be used as a weapon to procure manuscript revisions.

`\readingapproval` Uses text set up by preamble commands *thesistype, author, committeechair, departmentchair, graduatedean.* Produces a **Final Reading Approval** form including signature lines. This form is addressed to the Graduate Council and certifies that the thesis has been read and approved. Signatures are generally postponed until all revisions have been made and the committee approval form is completed.

`\prefacesection{Abstract}` Creates a separate abstract. The abstract in this case is set up with the command `\prefacesection{Abstract}`, then your abstract text follows normally. Figures, references, and such are not allowed. Don't use this one *and* the next one: use only one or the other.

`\preface{Abstract}{filename}` Creates the abstract, getting the contents from the file `filename`. This is useful if the abstract file is to be used by something else at the same time.

`\dedicationpage` Uses the text set up by the preamble command `\dedication` to create a dedication page.

`\tableofcontents` Uses text set up by sectioning commands. Part, Chapter and Section commands cause automatic entries in the `toc` file. This file is made by a pass through LaTeX. It will take two passes to get it right (with no intervening aborts). On each pass, the last table of contents is inserted into the output, and a revised table of contents is written onto the `toc` file.

The table of contents might need extra blank lines and additional text. The command which does this is called

```
\addcontentsline{#1}{#2}{#3}
```

For this purpose, #1 always equals `toc`. The allowed values of #2 are: *part, chapter, section, subsection, groupheader.* The *appendix* uses *chapter*, so **appendix** is not an allowed keyword.

Use #2 equal to *groupheader* to insert text without page references and dot leaders. The value of #3 is any text you want to put on the line. For example,

```
\addcontentsline{toc}{groupheader}{\protect\newline}
```

adds a blank line into the TOC (vertical space). Another useful idea:

```
\addcontentsline{toc}{groupheader}{\vspace{-1pc}}
```

which subtracts 12pts of vertical white space from the `toc` file. A macro `oneline` written for this purpose is:

```
\def\oneline{\addcontentsline{toc}{groupheader}{\vspace{-1pc}}}
```

See page 159 of the LaTeX manual.

`\listoftables` Uses text set up by table environment captions. The list is single-spaced with double-spacing between entries. Table titles are supposed to agree with the first sentence used in the caption, regardless of the complexity of the table caption. The formatting of TOC entry and title may be different. The LaTeX manual suggests 100 characters maximum for the list of tables entry and 300 characters maximum for the table title. See the section below on figure and table placement for details about captions.

An entry in the list of tables is created by the `\caption` control, provided the control `\captionONtrue` is set (the default). The setting of `\captionONfalse` caused the `\caption` control to abort incrementing any counters and no entry is made in the list of tables.

Figure and table captions. Please do not confuse figure captions and table captions. They are different. A figure caption might be a narrative, but a table caption is usually restricted to one or two lines, in general. Anything to be said about the table and its data is either in the text or else is a paragraph *below* the table. The SIAM journals tend to have table captions in paragraphs. So, it is *allowed* to follow a SIAM paragraph style for tables. How you set the style of tables depends a lot upon the journal where you hope to publish. See journals in your specialty and the standard style guides for examples.

`\listoffigures` Uses text set up by figure environments. List is single-spaced with double-spacing between entries. The best titles are one line long or less. Follow the recommendations above for table titles and list of tables entries. In particular, read above about the `caption` control and the two boolean controls `captionONtrue` and `captionONfalse`.

`\preface{Acknowledgments}{filename}` Generates a preface page, using the text written in the file called `filename`. The contents of this LaTeX source file is up to the author. A *prefacesection* command can do the same job, if the text is included directly.

`\optionalfront{name}` Additional sections may be added using this command. The `name` is the name of the section, such as “Glossary”, “List of Symbols”, or “List of Abbreviations”. This command just sets up the page heading and adds an entry into the table of contents. The format of the section is up to you, and it is advisable first to follow the general format of the other front matter sections, and then to check with the thesis editor before trying to get Format Approval.

5 Vertical space controls

A common problem in a thesis is to control the vertical white space in a way that can be easily undone. It is a mistake to use the common controls of LaTeX, because they cannot be turned off without affecting the internal workings of the typesetter.

Below are control sequences that can be inserted into the document to add and subtract vertical white space. If the document is later used to produce a journal article, then the controls can be disabled by a few lines of definitions in the preamble. For example, a relevant control `\foo` can be easily turned off with `\def\foo{}`. Not so with `\vspace{...}` commands, because they are used internally also, and cannot safely be turned off.

`\minusfourthline` Subtracts vertical space equal to one fourth of a blank line. The definition:

```
\def\minushalfline{\vspace{-0.25\normalbaselineskip}}
```

`\minushalfline` Subtracts vertical space equal to one half of a blank line. The definition:

```
\def\minushalfline{\vspace{-0.5\normalbaselineskip}}
```

`\minusline` Subtracts vertical space equal to one blank line. The definition:

```
\def\minusline{\vspace{-\normalbaselineskip}}
```

`\plusfourthline` Adds vertical space equal to one fourth of a blank line. The definition:

```
\def\plushalfline{\vspace{0.25\normalbaselineskip}}
```

`\plushalfline` Adds vertical space equal to one half of a blank line. The definition:

```
\def\plushalfline{\vspace{0.5\normalbaselineskip}}
```

`\plusline` Adds vertical space equal to one blank line. The definition:

```
\def\plusline{\vspace{\normalbaselineskip}}
```

`\captionONfalse` This boolean causes the `caption` macro to turn off incrementing of the table and figure counters and it causes each use of `caption` to abort producing an entry in the list of figures or the list of tables. Generally used before a continuation table.

`\captionONtrue` The default. This boolean causes the `caption` macro to turn on incrementing of the table and figure counters and it causes each use of `caption` to produce an entry in the list of figures or the list of tables, as appropriate.

`\captionlineskip` The spacing of lines in table and figure captions is controlled by the dimension definition

```
\def\captionlineskip{13.5pt}
```

which is the standard for single spacing. To change the spacing, specify the dimension, for example

```
\def\captionlineskip{14.5pt}
```

could be used to control spacing in a caption that contains many uppercase keywords. The control can be set and reset many times, once for each caption, if needed. Be aware that consistency is required in captions: any major change is expected to be uniform throughout the thesis.

`\mainheadingwidth` The horizontal width in a heading for `part`, `chapter`, `section` commands is controlled by the dimension definition

```
\def\mainheadingwidth{4.25in}
```

To change the spacing increase or decrease the value 4.25, for example:

```
\def\mainheadingwidth{4.5in}
```

The control can be set and reset many times, once for each main heading, if needed. This may happen if the title contains a large math symbol or the breaking of a title into inverted pyramids fails to work properly.

This dimension is also used for headings in the front matter. It is suggested that you change the dimension only after `\maintext` in the LaTeX source. After this point, it can be reset at will.

Fixing errors in sectional titles. Long titles for all sectional commands except `part`, `chapter` and `paragraph` produce an error message if the title exceeds the width limit:

```
Title Error:  (62.04767pt too wide)
Page 1, Title="Second section:  consequences of the first section"
```

This can be turned off by `\noisyfalse`. Titles are checked for dimension `\mainheadingwidth` less half an inch for the section number. The maximum title size is 4.5in including the numbering. A little more space (4.8 or 4.9) can be used for a particular title in order to make it look good.

Part and Chapter titles are set in a box of width defined by the control `\mainheadingwidth`. Error messages are automatically emitted for exceeding the box width. The setting of `\noisyfalse` does not affect these error messages.

Long title problems can be fixed individually. To produce 5pt more vertical space on the second line of a long title, the line break can be coded `\protect\\[5pt]`. All line breaks should be coded `\protect\\` and not simply as `\\`. It is good practice to define a control `\def\BR{\protect\\}` and then use `\BR` in all titles to insert a break.

Fixing errors in the table of contents. Table of contents entries sometimes have to be supplied with breaks and separate entries for the TOC and the title in the text. Normally, the TOC is set with entries up to 4.5in wide, with breaks supplied by LaTeX. For example:

```
\chapter[The Longest Day of the War]
        {The Longest Day \protect\\ of the War}
```

The errors in title lengths for table of content entries are emitted by LaTeX during the input of the `toc` file. These are *serious messages* which direct you to fix the `toc` entry that originates from a sectioning command. For example, the LaTeX error message

```
[5] (thesis.toc
Overfull \hbox (6.62102pt too wide) in paragraph at lines 16--16
 \twlrm sec-tion, first sub-sub-sub-sec-tion, first para-graph:
)
```

means that some sectioning command title had a bad break in the TOC. Furthermore, the bad break was on the word `paragraph`. Insert a manual break to get it right. Find the title in the source and do something like this (`\BR` is defined by `\def\BR{\protect\\}`):

```
\paragraph[Third section, third subsection, first subsubsection,
first subsubsubsection, first \BR paragraph: must be a sentence.]
{Third section, third subsection, first subsubsection, first
subsubsubsection, first paragraph: must be a sentence.}
```

`\doublespacedheadings` The thesis format requires every other line to be blank throughout the thesis. Chapter titles are included, they must have every other line blank. Titles of sections, subsections and subsubsections must have a single style. The *first style* is every other line blank (double-spaced titles). It is an option that can be specifically selected with the above control sequence.

`\singlespacedheadings` The *second style* is text on every line and no interleaved blank lines (single-spaced titles). This is used only for the `subsection`, `subsubsection`, `subsubsubsection` commands. The others remain the same as in the `doublespacedheadings` format. This option is the default as a result of consultation with the thesis office (1993).

`\fixchapterheading` The space after a main heading is too large if a chapter command is followed by a section command with no text between. Use `\fixchapterheading` to remove the extra space. Place this command after the `chapter` command and before the `section` command. This command applies also to an appendix section that directly follows an appendix heading (chapter followed by section in an appendix). The default definitions for single and double spaced headings:

```
\def\fixchapterheading{\vspace{-24pt}}
\def\fixchapterheading{\vspace{-19pt}}
```

6 Main Text

The chapters are to follow the above special page-generating commands, but to set the counters correctly, the command

```
\maintext
```

must precede the `\include{...}` or `\input{...}` commands for the chapters.

The spacing after a chapter command is considered to be a critical parameter by the thesis editor. A thesis should be coded in LaTeX as follows:

```
\chapter{Preliminaries}
\fixchapterheading             % Used when no text appears
\section{Introduction}
```

There is no way to predict the contents of the first line of text following a chapter command. It could be a sectional command or it could be a paragraph of text.

If you want a draft without the front matter pages, then comment out the front matter command or commands, and leave just the command `\maintext` instead. The draft will begin unceremoniously with the first page of the first chapter. It will still include all the back matter. If your thesis has a `part` command, then it should follow `maintext` so that the page numbering begins correctly (PART I is on page 1).

Each chapter or part should be in its own file, and included into the thesis using the LaTeX command `\include`. The command `\input` may also be used, but it is more suitable for direct textual inclusion, while `\include` has some additional capabilities that are useful while working on drafts. In particular, the `\includeonly` command is very useful to restrict processing to selected `\include` files—important when the whole thesis may take 30 minutes to format! A disadvantage of the `\include{...}` command is that each use clears the page. If the chapters are broken due to length problems, then extra white space may appear (so break on section or chapter headings only).

Part and chapter headings are defined by the thesis format and never change, but the exact format of section headings depends on how many levels of subheading you use. Since it would be pretty hard to decide by looking directly at the text, you will need a command `\threelevels`, `\fourlevels` or `\fivelevels` in the preamble (before the `begin{document}`).

7 Font sizes and styles for sectional commands

Section headings can be `part`, `chapter`, `section`, `subsection`, `subsubsection`, `subsubsubsection` or `paragraph`.

The `part` and `chapter` headings are the largest fonts used. The format of these headings is: maximum 4.5 inches wide, centered, every other line blank. The default size is 4.25in, changed with the control `\mainheadingwidth`.

The test for blank line height is made by placing two duplicate pages together in registration back lit by a strong light source. The two pages together must make a full page of text without collision of characters from adjacent lines. This test implies that blank line height depends upon the font height of adjacent lines, generally measured by uppercase `M`. This format is called *double-spaced*. Normal LaTeX spacing is called *single-spaced*.

The sectional commands `part` and `chapter` are always produce a double-spaced heading.

The commands `section`, `subsection`, `subsubsection` and `subsubsubsection` may be all single-spaced or all double-spaced (but not mixed). See the command `\doublespacedheadings` to select the non-default style.

The `section` command produce a number which is part of the first centered title line. Additional lines are also centered. Specific line breaks must be supplied by the author to achieve inverted pyramid format.

In `threelevels`, the `subsection` commands produces a number in its own box and the title is set in the selected font in a paragraph of width 4.25 inches. It is up to the author to set specific line breaks so that the title is left-justified and forms an inverted pyramid on the right (maximum width is 4.5in).

In `fourlevels`, the `subsection` is centered like `section`. The `subsubsection` appears as outlined above for the `subsection` control in `threelevels`.

In `fivelevels`, the `section` and `subsection` are centered. The `subsubsection` and`subsubsubsection` are essentially the same, with `subsubsection` being the same as `threelevels`.

In `threelevels`, `fourlevels` and `fivelevels`, the command `paragraph` produces a boldface sentence at the start of a paragraph. The text should always end in a period and form a sentence. A number is generated at the beginning of the line. The paragraph always appears in double-spaced text. Generally the first sentence should be less than 4.5in long (it will appear in the TOC).

A sectional command must be followed by two lines of text. That means a sectional command cannot end a page and clearly it should not split on a page boundary. There is no uuthesis intervention on placement, it is up to the author. The placement can be forced by writing `\vbox{...}` around the sectional command and the paragraph that follows.

The following scheme is used for assignment of font sizes and styles for double-spaced titles in sectional commands.

Command	Font size and style	Details
`part`	`\HFpart\bf`	uppercase title, inverted pyramid, doublespaced. Use whole page, centered horizontally and vertically.
`chapter`	`\HFchapter\bf`	Centered uppercase title in 4.25in box, double-spaced, inverted pyramid. Box can be up to 4.5in wide. Chapters are expected to start a page.
`section`	`\HFsection\bf`	Centered, dual-case title, doublespaced. Inverted pyramid up to 4.5in wide.
`subsection`	`\HFsubsection\bf`	Dual-case title, doublespaced, inverted pyramid up to 4.5in wide. For `fourlevels` and `fivelevels` the text is centered. For `threelevels` it is left-justified.
`subsubsection`	`\HFsubsubsection\bf`	Left justified, dual-case title. The subsubsection number uses 7 characters and the title is a doublespaced paragraph left-justified 10 characters from the edge. The number is separated from the title by 2 spaces. One blank line follows the title.
`subsubsubsection`	`\HFsubsubsubsection\bf`	Left justified, dual-case title. The subsubsubsection number uses 9 characters and the title is a doublespaced paragraph left-justified 13 characters from the edge. The number is separated from the title by 2 spaces. One blank line follows the title.
`paragraph`	`\HFparagraph\bf`	Standard paragraph indent, dual-case title. The paragraph number uses 9 characters. After 2 spaces the title is run into the doublespaced paragraph. Without the paragraph number this amounts to bold-facing the title at the start of a paragraph.

The 12pt font sizes are listed below. All titles are set in bold-face style. The sizes and styles can be changed with special permission from the thesis office.

`\HFmainhead`	Uppercase `\large`
`\HFpartHT`	Uppercase `\large`
`\HFchapterHT`	Uppercase `\large`
`\HFsectionHT`	Mixed case `\large`
`\HFsubsectionHT`	Mixed case `\normalsize`
`\HFsubsubsectionHT`	Mixed case `\normalsize`
`\HFsubsubsubsectionHT`	Mixed case `\normalsize`
`\HFparagraphHT`	Mixed case `\normalsize`

The baseline skips for double-spaced headings are invoked by the control `\doublespacedheadings`:

```
\def\HFmainheadHT{24pt}
\def\HFpartHT{24pt}
```

```
\def\HFchapterHT{24pt}
\def\HFsectionHT{20pt}
\def\HFsubsectionHT{20pt}
\def\HFsubsubsectionHT{20pt}
\def\HFsubsubsubsectionHT{20pt}
\def\HFparagraphHT{14.5pt}
```

For 12pt single-spaced headings, the current *default*, the following changes are made by the control sequence `\singlespacedheadings`:

```
\def\HFsectionHT{15.5pt}
\def\HFsubsectionHT{13.5pt}
\def\HFsubsubsectionHT{13.5pt}
\def\HFsubsubsubsectionHT{13.5pt}
```

A typical sectional font definition is as follows:

```
\def\HFchapter{\@setsize\HFchapter{\HFchapterHT}\xivpt\@xivpt}
```

This command defines `HFchapter` as a heading font size using 14pt pre-loaded LaTeX fonts. For example, `\HTchapter\bf` is used by the `\chapter` command to switch fonts for a chapter heading.

If you want something other than the two defaults supplied in uuthesis, then it is up to you to learn about the sectioning commands and the above font definitions. Install changes in your own private file called `thesis.sty` by copying the appropriate code from uuthesis.sty and entering the modifications. The file `thesis.sty` is read *after* the file `uuthesis.sty` and hence new definitions replace the standard definitions. Simple changes like `\bf` to `\rm` should be easy. For font size changes, be prepared to show sample output and be able to justify why your thesis should be different.

8 Back Matter

The back matter of a thesis consists of the pages that follow the chapters. These pages consist of one or more appendices, a bibliography, perhaps an index and vita.

Appendices. The control `\appendix` applies in the usual way, as documented in the LaTeX manual. To get the formatting right for the number of appendices, set the variable `\numberofappendices=n`, where `n` can range from 0 up to any value. The style of the table of contents and the chapter headings will be different for 1 appendix than for 2 appendices.

Appendices can have sections and subsections. These will not appear in the table of contents.

```
\numberofappendices=2         % "Appendices" appears in TOC
\appendix                     % Switch into appendix mode
\chapter{Fortran code}        % appendix "A.1 Fortran code" in TOC
\fixchapterheading            % Fix spacing
```

```
\section{Common variables}      % Not written to TOC.
\section{Subroutine init}       % Not in TOC
\chapter{Numerics}              % appendix "A.2 Numerics" in TOC
```

The lower order sectional commands like `section` will not generate entries in the TOC. Larger subheadings in the appendix that need no TOC entry can use `\chapter*{...}`.

Presently there can be only one set of appendices by the rules of the graduate school. To make the controls complete, there is `\noappendix` to shut off the appendix mechanism and start chapters again where the numbering left off. The control records how to shut off the appendix mechanism, in case that documentation is required. This could become necessary for a thesis that has several parts, because the appendices in that case have a slightly different structure.

Bibliographies. There are at least two ways to do bibliographies — manually, or using BibTEX. The latter is a far superior alternative, especially if you intend to use the same references in several places (such as the papers you're going to extract from the thesis). Significant bibliographies may already exist for the references you want. For instance, the PASS group in Computer Science has a file with about 1000 bib entries, mostly in programming languages. Another advantage of BibTEX is that it will format the entries according to a predefined style declared by the command `\bibliographystyle`.

To date, there is no universal mathematics database. BibTEX use by mathematicians has been enthusiastic. Most theses in mathematics now do BibTEX bibliographies. Most researchers have access to large databases of bibliographic data for their specialty, largely compiled by individual effort.

Many bibliography styles have been defined. Computer Science people should use the `acm` style, which follows the ACM journal style. There is no universal style file for mathematics theses. A number of mathematics theses have used the `plain` and `SIAM` styles. Several Computer Science theses have successfully used the ACM style file. The ACM style uses lowercase article titles. Proper names are protected by using braces in the title.

`\rawbibliographytrue`
This control defines the bibliography environment suitable for placing references after a section title, as is done in the Chemistry Department. The section used is the last of a chapter and each chapter collects its own set of references, appearing in a numbered section, which also appears in the Table of Contents. Place this control in the preamble of the LATEX document to enable the feature. When enabled, the `bibliography` environment only creates a list of references, no headers or table of contents entries are generated (the section heading does this already).

`\rawbibliographyfalse`
This control defines the bibliography environment suitable for placing references at the end of the thesis, as is done in mathematics. The Table of Contents has the word "REFERENCES". The references appear on a new page with 2-inch main heading. This is the default. The control can be used many times within the thesis to control the output of a page of references.

Index. The thesis format allows for an index. Its format should be like that in the *Handbook* from the thesis office, which is basically the same as the standard LATEX behavior.

An index is a separate LATEX source file which produces a typical index with page references, not different from an index found in a textbook. The LaTeX source for the index might reside in a file `thesis.ind`. The index is inserted into the thesis root file by the following code (the end of the root file):

```
\input{thesis.ind}
```

One mystery is how to select the items to index. A second mystery is how to automate the matching of the page numbers to insure that the index is always correct. The mysteries are solved below.

Support within LaTeX for an index is restricted to *earmarking words and phrases* for output to a low-level file with extension `idx`. For example, `thesis.idx` is made from root file `thesis.tex` by placing `\makeindex` in the preamble and running LaTeX. The file `thesis.idx` contains all the earmarked words and phrases plus their page number in the text. This is a major service, because it solves the problem of matching page numbers to index items. It also insures that page number references to the index are correct. Unfortunately, the contents of `thesis.idx` are unsuitable for use as an index *directly*, because the list is unsorted and unformatted.

Words and phrases are earmarked in the document using the control `\index{#1}`. Nelson Beebe recommends using a control `\X{#1}` defined by `\def\X#1{{#1}\index{{#1}}}` because it is easier to tag the words and phrases. The `\index{...}` command produces itself no text in the document, only in the `idx` file. Actual entries in the `idx` file depend upon definitions of `\item`, `\subitem` and `\subsubitem` given in `uuthesis.sty`.

The low-level `thesis.idx` source file is as primitive as an `aux` file. It is made by inserting the control `\makeindex` into the preamble (and only there) of the TeX document.

The second mystery, that of sorting and formatting the index, will now be discussed. The creation of the sorted and formatted `thesis.ind` from the LaTeX low-level file `thesis.idx` is done in a terminal window outside of LaTeX (and also outside of `emacs`). We run a special unix command `makeindex` or `makeidx` to create the source `thesis.ind` from `thesis.idx`. The syntax for the command is given in a manual page: `man makeindex`. The 1993 version runs like this:

```
makeindex thesis.idx
```

It produces `thesis.ind` which is the sorted and formatted version of the index.

The formatting of the index itself for uuthesis style is two-column newspaper format with header **Index** and normal numbering. If you are going to insert an index, it must be before the *vita* and after the *appendices*. Because of numbering problems it may be necessary to turn off the appendix switch with `\noappendix`.

The special separate executable system program called `makeindex` or `makeidx` exists on SUN and DEC unix systems. It is also available for microcomputers and other operating systems. Contact Nelson Beebe for details about other operating systems, or look in the directories `pub/tex/pub/makeindex`, `pub/ibmpc` and `pub/macintosh` via anonymous ftp to site `science.utah.edu`

Vita. The vita is optional, but the absolute last thing if it appears. It is set up by the `\vita` command, which starts a new page and titles it. The text for your vita is to immediately follow the `\vita` command. It must follow LaTeX rules, but it is otherwise totally up to you. The Thesis *Handbook* makes some suggestions.

9 Figure and Table Placement

There is deep problem with figures and tables. They trouble most persons who use LaTeX. The placement seems sometimes out of control, with the figure or table ending up on the wrong page. The purpose of this section is to discuss how to adjust the placement and tweak various parameters to make everything happen according to plan.

The thesis format dictates that figures *should be placed so they look balanced on the page.* Defined limits: If text appears, then it makes a block of six or more lines mid-page. Pages of floats are not constrained. Figures and tables are separated by triple spacing (2ex). The counting of figures and tables refers to large boxes that take the full text width. A figure that itself consists of three small horizontally placed graphs is considered one figure.

The basic setup for figures and tables is to define them as floating boxes of text that take up the full text width. As such, a box can be placed at the top `[t]`, at the bottom `[b]` or in the middle of the page `[h]`. Figures and tables can appear by themselves on a page of floats. This is a page on which no text appears. The option `[p]` selects this placement.

The graduate school style disallows a figure mid-page, that is, text cannot appear both before and after a figure on the same page. A chapter can end with a figure. If too little of the page is used it may look better placed with `[h]` option as the last item on the page.

Further, a figure must appear *after* it is referenced, that is, the figure is physically below the reference, and no later than the next page, or the one after that, in case there is a page of floats generated. In summary, the thesis office allows options `[b]`, `[t]`, `[p]`. However, `[b]` and `[t]` have restrictions: `[t]` cannot generate a figure before the figure is referenced, and `[b]` cannot generate a figure on the following pages after the reference. The `[h]` option is used rarely to force a figure placement at the end of a chapter. It should be mentioned that at chapter end this is the *only way* to get the figure in the right place on the page!

The figure and table counters are reset in each chapter. Default placement is `[p]`. Figure and table numbers look like `2.5`, where `2` is the chapter number and `5` is the figure number in that chapter. The figure captions use **Figure 2.5** followed by the **Title** which is in the normal Roman 12pt font. The numbers and titles are inserted by the `\caption` control using a special syntax to generate cross reference labels. If the `caption` control is not used, then no counters are incremented and all labeling and numbering is up to you.

```
\newcounter{figure}[chapter]                  figure counter
\def\fps@figure{p}                            Default placement [p]
\def\thefigure{\thechapter.\@arabic\c@figure}   Number is "2.5"
\def\fnum@figure{\bf Figure \thefigure}       Label is "Figure 2.5"

\newcounter{table}[chapter]                   table counter
\def\fps@table{p}                             Default placement [p]
\def\thetable{\thechapter.\@arabic\c@table}    Number is "2.5"
\def\fnum@table{\bf Table \thetable}          Label is "Table 2.5"
```

For example, if you don't like the bold label, then insert this into your private `thesis.sty`:

```
\def\fnum@figure{Figure \thefigure}        Label is "Figure 2.5"
\def\fnum@table{Table \thetable}           Label is "Table 2.5"
```

The secret to getting figures and tables to dump in the right place is a function of several delicate parameters. They are described more fully in the LaTeX manual. There is no discussion of the effect of change of *two* parameters at once. To understand the full effect requires a deep knowledge of the float algorithm in `latex.tex`. Suffice it to say that the parameters below, set as indicated, cause the figures to be dumped in a predictable way. Still, it may not produce what you want, and hence you have license to change them locally mid-document to handle a particular positioning problem. *Exercise great caution* in changing these parameters. After they produce the desired change, *reset them to the defaults below*.

```
\def\topfraction{1}            % Option [t] can use the whole page
\def\bottomfraction{1}         % Option [b] can use the whole page
\def\textfraction{0}           % Page can be all figures
\def\floatpagefraction{.5}     % Float page dump if half full
```

The maximum number of figures on a page is controlled by some variables that you will likely never have to reset. Option `[t]` and option `[b]` can be used together to generate 20 figures on one page. The thesis office has no rule about the number of figures per page. Manual intervention is required. The variables and default values (don't change):

```
\setcounter{topnumber}{10}      % Option [t] can place 10 figures max
\setcounter{bottomnumber}{10}   % Option [b] can place 10 figures max
\setcounter{totalnumber}{20}    % Up to 20 figures per page
```

The distance between figures and text and tables is set by a number of glue variables. It is possible to change them locally within the thesis to solve fitting problems with figures and tables. Always reset the values after use!

```
\floatsep 2em plus 2pt minus 4pt        % vertical space between floats
\textfloatsep 2em plus 2pt minus 4pt    % vertical space between text and floats
\intextsep 2em plus 4pt minus 4pt       % vertical space around mid-page float
```

Figure captions. Captions for tables and figures are centered in the current text width. A figure caption is *below* the figure. A table caption is *above* the table. The captions are to be consistent and follow a recognized style guide. If a style guide is not followed, then the thesis office asks that a caption not extend beyond the table or figure edges and that the title be set in a centered inverted pyramid style. Rules for tables are different than rules for figures. See your style guide or find journal samples that support your style.

The captions in figures may be of a different style than found in style manuals such as the *ACS Manual* or the *Chicago Manual*. In this style, the caption is a paragraph set in block format across the text width. The title can be set in boldface font, followed by a paragraph of table notes.

The *Journal of Computational Physics*, Vol 108, No 1, September 1993, p 173, shows a non-pyramid form table with a paragraph of text instead of a short title. The *SIAM Journal of Scientific Computing* shows similar formats for tables in the July 1993 issue, Vol 14, pages 930-931. The *SIAM Journal of Numerical Analysis* Volume 30, No 3, June, 1993, page 625, shows very long table captions in block format.

A common style borrowed from many style manuals is in use by journals such as *Applied Numerical Mathematics*, *Journal of Combinatorial Theory*, *Journal of Number theory* and *Journal of Multivariate Analysis*. In this style, the title is above the table in a centered inverted pyramid format. The title is brief. If notes about the table are needed, then they appear below the table in a text paragraph or else in the main text near the reference to the figure.

The control

```
\captionstyleparagraph
```

sets the caption in a text width paragraph. The size can be changed with a `parbox` construct or a `minipage` environment wrapped around the caption.

The control

```
\captionstylecentered
```

sets the caption in a centering environment using the current text width. Again, the width can be changed by `parbox` and `minipage` constructions.

The text below a table, or table notes, as they are sometimes called, is set in a text width paragraph. This takes no special command inside a `table` environment, except possibly a `\par` command to insert space after the last line of the table. In particular, this text is *not* defined by a `caption` command!

The title used in a caption must also appear in the **List of Tables** or the **List of Figures** with the same words. The breaking of long titles will be different in the *List*. For example, in the Robert Dillon thesis, table entries and figure entries in the *Lists* were the first sentence in the paragraph that made up the caption. Caption placement is *below* the figure and *above* the table.

The controls `captionONtrue` and `captionONfalse` turn on and off, respectively, incrementing of counters and list of tables of list of figures entries, for usage of the `caption` macro. The best use of this feature is to create a continuation table on the next page which has the same table number and title *Continued*. All the features of LaTeX floats are available to place the table on the next page.

A figure caption may be a paragraph of text describing the table. The caption is expected to be in paragraph format or else in centered inverted pyramid format. Titles wider than the text width are unacceptable. The entry in the list of figures is expected to be in a different format, namely a plain text paragraph, and one line is best.

A table caption is usually not a paragraph, but in certain styles that also is possible. In paragraph style, the first sentence is kept short so it can be the same in the *List*. Explanation of the table is to appear *below* the table in a text width paragraph, or in the main text, or if you use the style of the *SIAM Journal of Numerical Analysis*, as a paragraph or less continuing the title. The keynote here is consistency: don't mix styles, use one style. Exceptions to this rule for tables are an agreement between the thesis office, the student and the thesis advisor. A style guide is the best way to procure agreement.

An example follows showing how to control the effects for a table of unspecified width of about 3in having a two-line title that is to be set in centered inverted pyramid format (ACS style). The attempt here is to constrain all formatting to the table width. It is more common to use the entire text width for the table with horizontal line across the page following the table. If that style is preferred, then replace `\TMPsize` by `\textwidth` in the example (so that footnotes appear below the table).

```
\captionstylecentered
\def\TMP{\begin{tabular}{|c|c|c|c|c|c|c|c|} \hline
 $x$ & 0.00 & 0.25 & 0.50 & 0.75 & 1.00 & 1.50 & 2.00 \\ \hline
 $y$ & 1.10 & 1.08 & 1.00 & 0.86 & 0.66 & 0.35 & 0.27 \\ \hline
\end{tabular}}
\newdimen\TMPsize\settowidth{\TMPsize}{\TMP}

\begin{table}[b]
\begin{center}
\begin{minipage}{\TMPsize}
\caption{Principal Coordinates for the \protect\\ Fundamental
Dynamical System}
\TMP\medskip
\par\rule{\hsize}{1pt}\par
\leftline{\footnotesize The value $x$ is the position coordinate.}
\leftline{\footnotesize The value $y$ is the velocity coordinate.}
\end{minipage}
\end{center}
\end{table}
```

In this example, *Table 2.15. Principal Coordinates for the Fundamental Dynamical System* will appear in the list of tables, while an inverted pyramid title will appear above the table, as follows:

Figure 2.15. Principal Coordinates for the
Fundamental Dynamical System

x	0.00	0.25	0.50	0.75	1.00	1.50	2.00
y	1.10	1.08	1.00	0.86	0.66	0.35	0.27

The value x is the position coordinate.
The value y is the velocity coordinate.

The **minipage** construct is used to restrain the footnote width and the caption width to the table width. The full text width is used by default for the caption. Explicit line breaks were needed to produce an inverse pyramid.

The best style uses the caption macro with a short title so that no engineering needs to take place in the list of tables.

Check list for figures and tables

1. A figure must follow its reference, at the bottom of the current page or immediately on the top of the next page. Start with `[b]` and change to `[t]` if it doesn't fit.

2. An option of `[t]`, which causes proper figure placement after the reference on the preceding page, can be followed by a figure with [`[b]` option. Both figures can be used on the same page. Warning: Three usually won't fit and still have six lines of text mid-page.

3. Three figures in a row should generate one figure at the bottom and two figures on the next page. This means one [b] option followed by two [t] options. The deferred figures may have to be physically moved in the source to cause the dumping onto the next page. Often the text looks best if the two deferred figures are on a page of floats by themselves. Use [p] on the last two figures to make it happen. The decision is based on six lines of text. If six lines won't fit on the same page as the two figures (at the bottom), then make a page of floats for the two figures.

4. If a page of floats has a widow figure at the end, then it is to appear as a [t] option on the next page of text.

5. A `\clearpage` command is sometimes needed to empty pages of floats before the next paragraph is read. The required `\clearpage` is inserted after the final figure.

6. At the end of a chapter it is sometimes desired to place a figure as the last addition to the last page. The last `\begin{figure}[b]` ... `\end{figure}` will mysteriously appear on the next page. The required `\clearpage` is inserted after the figure to fix the problem, and [b] and [t] usually have to be changed to [h] if the page is not full.

   If a good portion of the last page is blank, then the final page might look better with a ragged lower edge, which can be obtained by using [h] instead of [b]. The thesis office endorses this exception to the mid-page figure rule.

7. Expect the thesis editor to move the figures and tables around to make the document look balanced. Expect that you will move figures also, as more text is added or subtracted. The flux ends when the thesis is approved and the final copies are delivered. A healthy attitude of expected change is more useful than an attitude of permanent location.

8. Check all figure pages for rule violations: caption, numbering, at least six lines of text or zero lines of text per figure page. Tables have one style, figures another. The style must be consistent: a violation is inverted pyramid title on one table and paragraph format title on another.

10 Making Figures

There are a number of methods for creating diagrams and figures for use in a LaTeX document. The choice of method depends upon the type of figure and the method of generation.

Photographs. If these are to be inserted into a thesis, then the photos must be approved well in advance of publication by the thesis office. There are copyright restrictions that may apply, which could prevent acceptance of the thesis.

A number of theses have used color photos. The photos are pasted onto the master copy and the page is copied on special bond paper in a color xerox machine. Bring your own paper and ask that it be used — if they won't, go somewhere else. It is important to understand the mechanics, because it simplifies the preparation of the paste-up master. Also, it is easier to plan the photo reproduction, knowing that only one copy will be lost to pasting. Color xerox copies cost less than an archival photographic print. The necessary bound copies are about 10. In some cases, too much detail is lost by xerographic methods and only the paste-up technique can be used.

Mounting of monochrome photos requires dry mount tissue. Rubber cement cannot be used in a library copy. Cibachrome color uses a special mounting adhesive and neither dry mount tissue nor rubber cement can be applied. In short, seek technical advice on photographs.

Discussed below is the method for paste-up of photographs with final copies produced by color xerography. It must be emphasized that this method applies only in limited cases where resolution loss is not important to the photograph.

The basic method for photos is to create a framed box smaller than the photo and write into the box important information about the photo, such as where it came from, the size, the negative number, the date it was taken, and other data. The boxed area will be pasted over and the information lost, but not to you: it remains in the LaTeX source. Suggestion: use rubber cement or special cement for color plus alignment tools to position the photo. A drafting table with T-square is helpful.

The framed box is placed into the text by a figure environment. In Nelson Beebe's Postscript macros there is feature for doing what appears below. After you fight the basic methods in LaTeX, as done below, Nelson's macros will make sense and you will be able to use them effectively. An example to illustrate what is possible directly from LaTeX:

```
\documentstyle[12pt]{article}
\setlength{\textwidth}{6.4in}
\begin{document}
\def\Xsize{4.5} \def\Ysize{3.5} % Photo size by ruler is 4.25 by 3.25
\def\Msize{3in} % Message size 3inches
\def\Xp{0.75}    % Put at X=0.75in = (\Xsize - \Msize)/2
\def\Yp{2.5}     % Put at Y=2.5in = \Ysize - \Xp
\def\PHOTO{
 \parbox{3in}{ \small\rm\medskip
   Taken 5-19-1981 at LBL. Nikon with strobe. f2.8 on Royal Pan.
   Negative 11 in 'LBL Trip' dated May 1981. I own it, no copyright
   troubles. \par\bigskip\par \centerline{\large\rm PHOTO 10a} }
}
\begin{figure}[b]
\begin{center}\fbox{\setlength{\unitlength}{1in}
\begin{picture}(\Xsize,\Ysize)\put(\Xp,\Yp){\PHOTO}\end{picture}}
\caption{High-Speed photograph of a bullet.}
\end{center}
\end{figure}
\end{document}
```

Paste-up Figures. Figures can be pasted up to make the final copy of a thesis. Still, it is handy to have LaTeX decide the figure positions, numbering and captions. The ideas used above for photographs can be applied to paste-up figures.

Computer-drawn Figures. The best unix tool is called `xfig`. It is a drawing program for X-windows that emits source code for LaTeX or PicTeX. The output can be inserted into LaTeX documents by an `\input{filename}` command inside a figure environment.

Taken 5-19-1981 at LBL. Nikon with strobe. f2.8 on Royal Pan. Negative 11 in 'LBL Trip' dated May 1981. I own it, no copyright troubles.

PHOTO 10a

Figure 1: High-Speed photograph of a bullet.

The program `xfig` is maintained by Brian Smith at Lawrence Berkeley Laboratory. The 1993 version is `xfig 2.1.6`, available on `export.lcs.mit.edu` in `/contrib/R5fixes/xfig-patches/`. It exists for unix machines, in particular, it runs on newer SUN hardware.

Commutative Diagrams. There are two packages that can be used to produce commutative diagrams and box-style diagrams involving lines and arrows and text. The first is called `diagram`. See the sample theses below for syntax. The second is called `xypic`, by Kris Rose, Copenhagen, Denmark. Both packages use special fonts to create the diagrams and they can be used in various flavors of TeX. This method of producing diagrams is preferred over using `xfig` because of versatility and ease of use. Unfortunately, there is a steep learning curve. A large investment of time is required to produce the first diagram.

Maple Graphs. Plots that can be generated by equations or differential equations or by easily programmed algorithms can be produced in `maple`. The plot output can be chosen to be `unixplot` or `Postscript`.

Postscript output can be used by the methods outlined by Nelson Beebe. See below for a reference to his long article on the subject of insertion of postscript figures into LaTeX documents.

The `unixplot` output has the advantage of being editable in `xfig`. The output file, usually called `plotoutfile`, is filtered twice and passed into `xfig` for editing of labels, axes and extra things like shading and captioning.

To make a maple figure proceed as follows. First, plot a curve like $y = \sin(x)$. The maple V.2 command is `plot(sin(x),x=0..Pi)`. Save it as a `unixplot` file called `plotoutfile`. Translate out the maple errors in `plotoutfile` with a unix script called `unixplot.sh`. Then apply the filter `xfig2pictex` to `plotoutfile` (.xfig extension assumed) to produce file `plotoutfile.pictex`. [4]

The output is a coded source in LaTeX macros that is suited for use with a macro package called `pictex`, made for generating figures in LaTeX by drawing a pixel at a time. It is slow but accurate and produces figures directly into the DVI file, hence the figures can be viewed in any good DVI screen viewer. The use of `pictex` macros is free. The manual costs money, but with `xfig` no programming ability in `pictex` is required.

The resulting pictex file is inserted into a LaTeX document as follows:

```
Preamble:   \input{pictex.sty} % also works as a style file option

\begin{figure}[b]
  \Figure {The function $y=\sin(x)$.} {plotoutfile.pictex}
\end{figure}
```

The command `Figure` itself is defined below. Figure size is controlled by the commands in the unix shell source `xfig2pictex.sh`. Hand edit of the pictex file is possible to shrink or blow up the figure. Figure size changes can also be done inside xfig.

```
\newcommand{\Figure}[2]
% #1=title
% #2=pictex file name
{% begin command \Figure
    \protect\begin{center}%
    \input{#2}%
    \protect\end{center}%
    \protect\caption{#1}%
}% end of command \Figure
```

The editing features of `xfig`, which adds LaTeX commands over the top of figures, is invaluable. After getting used to the environment your productivity should increase and the quality of your figures should better than ever before.

Postscript Figures. It is possible to directly insert Postscript figures into LaTeX documents. Many microcomputer programs produce Postscript output and therefore it may be an advantage to insert this type of figure directly. Some unix programs which can create the postscript images: `gnuplot, xfig, idraw, freehand, matlab, xmath, maple` and `mathematica`.

Many theses in mathematics have successfully used postscript figure insertions. If there is a standard, then possibly it is postscript. This standard is not without problems. The biggest trouble is the size of labels and the pen size for plotting. Letters have to be 2mm high to pass the microfilm test (less than 2mm will vanish on the microfilm). The xerox test is applied to the pen size: it must not lighten or vanish upon

[4] In you don't have access to these tools send mail to `gustafson@math.utah.edu`. Sources are in the directory `/u/ma/gustafson/tmp/plot2fig`. The files are `unixplotfix`, `xfig2pictex.sh` and `unixplot.sh`.

being Xeroxed. If your figures are Postscript, then choose a large point size for the fonts to insure that reduction leaves the letters 2mm high. For example, 14pt `\large` fonts reduced 50% become 7pt fonts, which is almost exactly 2mm high.

Editing and addition of LaTeX labels over the top of postscript figures can be tiresome. It is not impossible: see PsFrag below. In simple cases addition of labels and text can be done by positioning of coordinates. A `\put` command in a `picture` environment will place the LaTeX text where you want it.

A ruler and test grid are helpful. It is also possible to do this on a CRT screen in X-windows. The idea is to make a special grid on A4 transparency material. Position the grid on a `ghostscript` or `dvips` display of the graphic, then write down the coordinates where the text is to be placed. The transparency is a *2-dimensional ruler* made especially for your screen size and the given display program.

Nelson Beebe has written an extensive article on insertion of Postscript figures, called *TeX and Encapsulated Postscript*. Since this article is published and also updated periodically with new ideas, it is best to contact Nelson at `beebe@math.utah.edu`. The 1991 version 0.38 of the article was 44 pages (with index and figures).

Nelson recommends reading first the short article (5 pages, 9 figures) of December 1992 called *Incorporating Encapsulated PostScript in TeX and LaTeX Documents*, available as `exepsf.ps` in math directory `/u/cl/doc` or email a request to Nelson, `beebe@math.utah.edu`.

NCAR Graphics. The graphs produced by NCAR can be exported to Postscript format. The remarks of Nelson Beebe in his article on Postscript Figures apply to the exported figure files. See above for information on that article. Presently, there is no export of NCAR graphics files that can be filtered into `xfig`. Therefore, editing of the graphics is limited to available tools for postscript figures.

Editing Postscript Figures. There was an announcement of editing tools for postscript sources by

> Craig Barratt, `craig@ISL.Stanford.EDU`.

The package `PsFrag` is available via anonymous ftp from `isl.stanford.edu` (internet address 36.60.0.10). Get the compressed tar file `pub/boyd/psfrag/psfrag.tar.Z`. Extract the archive using `pdtar xzf`. See the files `README, USAGE, INSTALL` for detailed information. The package is a set of shell scripts that employ `dvips` and `ghostscript` to replace postscript text items by LaTeX code that overlay the postscript image.

Gnuplot Figures. The program `gnuplot` can produce publication quality figures with either LaTeX output or `xfig` output. The latter is best if you want to make a quick figure and then edit it in `xfig` to add labels, captions and change things like the line size, axes and fonts. Everything that `gnuplot` produces can be edited manually in `xfig`.

There is a document available on `gnuplot` figures called *LaTeX and the GNUPLOT Plotting Program* by David Kotz, `dfk@cs.duke.edu`, February 10, 1990. The LaTeX file that produces the documentation is available at a number of sites on internet. Send mail to `info-gnuplot@dartmouth.edu` (from version 3.5, 1993). This document is included in the `Tutorial` directory of the gnuplot distribution.

Graphs created by gnuplot can be exported in LaTeX format for direct inclusion in a figure environment. For example, if you want $y = \sin(x)$ in your thesis try the following:

```
(terminal window): gnuplot
```

```
(in gnuplot):
                set terminal latex
                set output "sin.tex"
                plot sin(x)
                quit
```

The resulting file `sin.tex` can be inserted into a figure environment with the command `\input{sin.tex}`. The figure size can be changed by editing `sin.tex` on the line containing `unitlength`.

Gnuplot figures can be produced from raw data files of (x, y)-values. Consider the following command file `fig1.2.1.gnuplot` invoked as

```
gnuplot fig1.2.1.gnuplot
```

in a terminal window:

```
set nokey
set data style lines
set tics out
set xrange [0:24]
set yrange [120000:240000]
set xtics 0,12
set ytics 120000,60000
plot "fig1.2.1.data"                        View result in X
pause -1                                    Wait for key
set term bfig                               Output large xfig file
set output "fig1.2.1.fig"
replot
set output                                  Back to X11 output
!xfig fig1.2.1.fig                          Run xfig and
fig2dev -L pictex fig1.2.1.fig fig1.2.1.pictex  Make pictex file
!emacs fig1.2.1.pictex fig1.2.1.gnuplot     Hand edit sources
```

Graphs from data using maple. The computer algebra system `maple` can make a figure from a given data file by writing the data into a list `L` and then invoking `plot([L])` to produce the graph. Many of the nice features of gnuplot are lost but others are gained. Virtually everything done in gnuplot above can be reproduced by a maple script. The real problem is that neither maple nor gnuplot provide any editing capabilities — a program like `xfig` is still needed along with translators and filters for import and export of the various graphics formats.

11 More Features

11.1 Additional Commands

Some additional convenience commands have been defined.

`\begin{epigraph}` *words* `\end{epigraph}`

Creates an epigraph. An epigraph is a quotation or motto at the beginning of a book, a chapter or section. The text will be indented and single-spaced, but the rest of the formatting (such as a blank line between the text and attribution) must be done by hand. Overdoing the number of epigraphs will cause you to be known as an *epigraphist*, one versed in *graffiti*.

`\begin{topics}{template} ... \end{topics}`

This special environment can be used to solve display problems that might otherwise use `tabular` or `array` constructions. It is similar to the `itemize`, `description` and `enumerate` environments which have the special feature of allowing page breaks between items. The definition:

```
\newcommand{\topicslabel}[1]{\mbox{#1}\hfil}
\newenvironment{topics}[1]{
    \begin{list}{}{%
      \let\makelabel\topicslabel
      \settowidth{\labelwidth}{\topicslabel{#1}}
      \setlength{\leftmargin}{1.1\labelwidth}}
}{\end{list}}
```

The special feature of the `topics` environment is seen in the example below. The alignment of items cannot be done in the usual list environments. But the new feature has a price: the longest item must be entered as a template at the start of the environment (similar to the `tabbing` environment). The template has a natural length that is used as the item width (see the above definition) and therefore custom widths can be designed for the item column which will persist and be uniform from page to page. The `\item[...]` is never optional: the square brackets must occur even if no text is to be entered!

```
\begin{topics}
      {\bf Crocodiles and Reptiles.}
\item[\bf How X works:]
   \begin{topics}{\sf Windows.}
   \item[\sf Windows.]
        Usually made of glass but
        X-windows are an exception.
   \item[\sf Errors.]
        They appear in the console
        window from time to time.
   \item[\sf Exit.]
        For the twm manager, exit
        by pull-down menu option.
   \end{topics}
 \item[\bf Alligators.]
      In many places in the world
      there can be found
      crocodiles and alligators.
 \item[\bf Crocodiles and Reptiles.]
      Besides crocodiles and alligators
      there are brown and green lizards.
\end{topics}
```

How X works:	Windows.	Usually made of glass but X-windows are an exception.
	Errors.	They appear in the console window from time to time.
	Exit.	For the twm manager, exit by pull-down menu option.
Alligators.	In many places in the world there can be found crocodiles and alligators.	
Crocodiles and Reptiles.	Besides the crocodiles and alligators there are brown and green lizards.	

`\noisytrue`, `\noisyfalse`

Enables and disables the printing of progress messages. This is useful for debugging if front matter is coming out wrong, or if you like programs to give out reassuring patter about what they're up to. The default is `\noisytrue`.

`\begin{singlespace}` *words* `\end{singlespace}`

This environment allows single-spaced text to be inserted into the thesis at any point. Use it with caution: the Thesis Manual is very specific about double-spaced text, which is the default in this style. A warning: a control `\par` is required before invoking single spacing and directly after as well. This can also be done by blank lines. Most complaints about single spacing can be resolved by inserting blank lines before and and the environment.

`\singlespace`
`\doublespace`
`\normalspace`

In some circumstances, the singlespacing environment may not work. It usually requires a `\par` control before it is used and immediately after before changing to other spacing. Each of these commands changes the inter-line spacing for everything up to the next space-setting command. `\singlespace` and `\doublespace` do the obvious things, while `\normalspace` sets to the default spacing (double for theses, single for tech reports). For this reason, `\normalspace` is preferable to `\doublespace`, when you only want to back to the usual spacing.

`\twopagefigure{#1}{#2}`

This special command is made for inserting photographic paste-ups; the first page gets the caption, and the second is completely blank. Its arguments are the figure caption and a label. The figure caption should contain a label also in order to build a list of figures entry. Its definition may be helpful to understanding how it is to be used:

```
\newcommand{\twopagefigure}[2]{
\begin{figure}[p]
\vspace{4in}
\caption{#1}
\label{#2}
\vspace{4in}
\end{figure}
\begin{figure}[p]
\vspace{8in}
\end{figure}
}
```

`\theoremsetup`

This setup control defines theorem-type environments `theorem`, `proposition` and `corollary` in a default manner. Put the control into the preamble of the document if you wish to enable the feature. Remove the reference to disable the loading of these default definitions. Warning: once defined, there is no convenient way to erase the definitions.

The reason for the control `\theoremsetup`, clumsy as it seems, is to allow for custom definitions of `theorem`, `proposition`, `corollary`. The `\newtheorem` feature disallows a definition to be turned off or re-defined. A common tweak of the theorem environment is to insert Roman text in place of italics into the theorem body. Of course, it is desirable to call the result `theorem` and not some other name.

`\begin{theorem}` *words* `\end{theorem}`

This theorem environment follows the suggestion in the LaTeX manual. It defines a new counter called `theorem` which is reset in each section and incremented upon each use of the environment. See the control `\theoremsetup` above to enable this environment. New environments can be made from a base theorem environment:

```
\newenvironment{Theorem}{\begin{theorem}\rm}{\end{theorem}}
```

This environment, used in place of `theorem`, has all the features of the theorem environment but sets the text body in Roman. The `Theorem` environment is *not defined* in the uuthesis style — it is a customization to be coded into the preamble.

`\begin{proposition}` *words* `\end{proposition}`

Like the `theorem` environment, this pre-defined environment follows the standard use of propositions and theorems, with the text of the proposition set in italics. The counter used is the `theorem` counter: there is no separate counter for propositions. See `\theoremsetup` above to enable this environment.

`\begin{corollary}` *words* `\end{corollary}`

Like the `theorem` environment, text is set in italics. The same counter is used: `theorem`. See the control `\theoremsetup` above to enable this environment.

Other inventions commonly used for a thesis can be included in the preamble of the document. Here are some normal definitions, which should be self-explanatory. Please observe that the items below are **not pre-defined**! To make them useful, enter them into the preamble of the LaTeX document.

```
\newtheorem{theorem}{Theorem}[chapter]          The common definitions
\newtheorem{proposition}{Proposition}[chapter]  enabled by control
\newtheorem{corollary}[proposition]{Corollary}     \theoremsetup

\newtheorem{lemma}[theorem]{Lemma}
\newtheorem{define}[theorem]{Definition}
\newtheorem{hope}[theorem]{Conjecture}
\newtheorem{example}[theorem]{Example}
\newtheorem{obs}[theorem]{Observation}

\newtheorem{lemma}{Lemma}[section]
\newtheorem{co}[lemma]{Corollary}
\newtheorem{df}[lemma]{Definition}
\newtheorem{example}[lemma]{Example}
\newtheorem{question}[lemma]{Open Question}
\newtheorem{remark}[lemma]{Remark}
\newtheorem{th}[lemma]{Theorem}

\renewcommand{\thelemma}{\arabic{section}.\arabic{lemma}}
\renewcommand{\theequation}{\arabic{section}.\arabic{equation}}
```

The font used for these environments is by default italic. If you want a normal roman font, then new environments must be defined — see the LaTeX manual for details.

`\begin{Proof} ... \end{Proof}`

The Proof environment is by Nelson Beebe. It uses normal Roman text, with a trailing black square at the end of the text, supplied automatically by the `\end{Proof}`. The definition:

```
\newcommand{\boxx}{\unskip \nopagebreak \hfill ${\blacksquare}$% }
\newenvironment{Proof}%
    {\noindent{\bf Proof.}\begin{rm}}%
    {\boxx\end{rm}}
```

`\proofline{#1}`

Insert a line of text used to identify a proof. It can be used also to invent a proof environment, which might be more convenient. The definition allows for easy re-definition:

```
\def\proofline#1{\protect\par\noindent{\bf #1}:~}
```

`\proof{#1}`

Inserts a proof line of the form "**Proof** `#1`:" which is convenient for items of the form **Proof of Theorem 7:**. The definition allows it to be redefined easily:

```
\def\proof#1{\proofline{Proof #1}}
```

`\pf`

Inserts **Proof:** into the text with no indent. This is a common way of marking proofs. The definition:

```
\def\pf{\protect\proofline{Proof}}
```

`\qed`

This control sequence is used to end a proof. It has the following definition in the uuthesis style:

```
\def\qed{\protect\par\noindent{}$\square$\protect\par}
```

Other possible definitions, placed in the preamble, would be:

```
\def\qed{\nobreak \hfill \rule{1ex}{1em}}
\def\qed{\nobreak\ \mbox{~~\rule{1ex}{1em}}}
```

`\mainheading{#1}`

The two-inch top margins that you see in the thesis are produced by this control sequence. It may be useful in solving some formatting problems. It is best used after an `\eject` or `\clearpage` command, so that the two inches of space is at the top of the page. The argument is the text that you want centered in `\large\bf` print, just below the 2-inch white space. This command actually uses a special font size `\HFmainhead` with baselineskip `\HFmainheadHT`, basically large boldface. The text is set in a box of width `\mainheadingwidth`, normally 4.25in.

`\ulabel{#1}{#2}`

This feature supplies an alternative to the `\label` control sequence. Its purpose is to prefix entries in the AUX file with a string given in the second argument `#2`. An example showing how `ulabel` works:

```
Example:
         \begin{theorem}% Theorem 2.4
         \ulabel{hottheorem}{Theorem}
          The group ${\cal G}$ is simple.
         \end{theorem}
```

Later on, \ref{hottheorem} produces string Theorem 2.4 because \thetheorem equals 2.4. The usual \label{hottheorem} supplied by LaTeX would produce only 2.4 in the AUX file and \ref{hottheorem} in this case would be 2.4 instead of Theorem 2.4.

Greg Conner used this control sequence in 1992 to reference theorems and corollaries. The advantage is that references can be tied to the particular theorem, corollary or lemma by words and number, resulting in a consistent system of references.

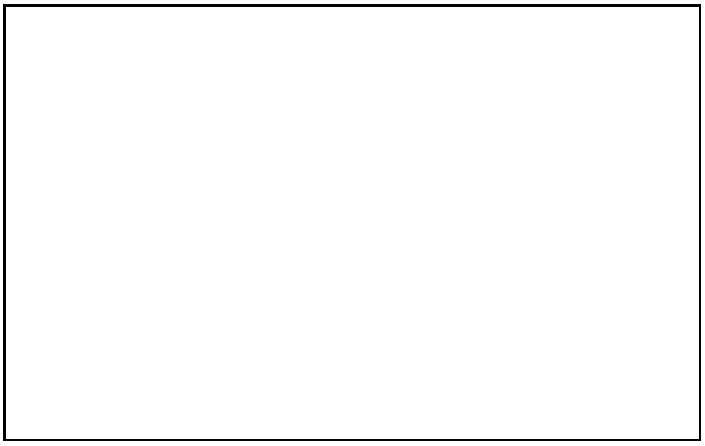
The definition:

```
\def\ulabel#1#2{\@bsphack\if@filesw {\let\thepage\relax
   \def\protect{\noexpand\noexpand\noexpand}%
   \xdef\@gtempa{\write\@auxout{\string
   \newlabel{#1}{{#2 \@currentlabel}{\thepage}}}}}\@gtempa
   \if@nobreak \ifvmode\nobreak\fi\fi\fi\@esphack}
```

\inputpicture

Jeff McGough used the control \inputpicture in November 1992 to input figure files in his thesis. He found that complicated figures slowed the evaluation process considerably and it became convenient to compile the DVI file with *phantom figures* in place of the real ones. The *phantom figure* was a box as seen below made with this LaTeX source:

```
\setlength{\unitlength}{0.0063in}%
\begin{picture}(573,357)(114,408)
\thicklines
\put(114,408){\framebox(573,357){}}
\end{picture}
```



The generation of figures is controlled by

```
\newcommand{\inputpicture}[1]{%
   \ifinputpicture \input{#1} \else \input{box.tex} \fi }
\newif\ifinputpicture \inputpicturetrue
```

where file box.tex contains the contents given above.

To use the \inputpicture control, enter after the beginning of the document one of these two commands:

```
\inputpicturetrue
\inputpicturefalse
```

To input a picture file `pic.tex` use the following syntax:

```
\inputpicture{pic.tex}
```

The picture will appear in the DVI file if the control sequence `\inputpicturetrue` was set and a box will appear in the DVI file if `\inputpicturefalse` was set.

11.2 Other Documents

The thesis itself is not the only document that you may need to produce. It is useful to have a file `forms.tex` which has all the declarations of the regular thesis, but only generates, say, the committee approval page, so it can be printed on bond paper and signed before the thesis goes through Format Approval.

It may be desirable to use a title page for the abstracts that are submitted separately; `\abstracttitlepage` works just like `\titlepage`, but produces a slightly different output.

Another possibility is for the thesis to be turned into a departmental technical report. This is required in Computer Science. However, the thesis format does not follow the report format, so as mentioned earlier, the `report` option to the style makes appropriate changes (there are really only a few that need to be made). The most significant change is the use of a different title page, which is generated by the command `\reporttitlepage`. The default definition of the command is specific to Computer Science, and it should be redefined if used elsewhere.

12 Warnings and Problems

The uuthesis style will not correct forms of references, spelling mistakes, or poor writing. Proofread everything.

A common complaint is inconsistency of use. An example: in the introduction the terminology **4–dimensional** used is whereas later on it becomes **four–dimensional**. While not a difficulty for American mathematicians, it is definitely a problem for persons whose native language is not English. A thesis is an international document, and therefore the manuscript must pass a consistency test of this nature. Nelson Beebe recommends using a control sequence for each piece of special terminology, to insure consistency throughout the document.

Double words are a common defect in theses. These can be eliminated by a unix program `dw` that checks for this situation. See the sample thesis sources in `/usr/local/lib/tex/uuthesis` on mathematics department unix machines or send email to Nelson Beebe.

Missing words are another common problem. Careful proofing of text is the only sure cure for such defects.

Placement of punctuation is determined by the style guide. It is to be consistent. Equations at the end of a sentence should end with a period, in general. Commas often appear at the end of a displayed equation.

Quotations should be treated as per the style guide: sometimes they get quote marks and sometimes not. Italics and periods are used according to the dictionary: is it 'et al.', '*et al.*', 'et. al.' or '*et. al.*? Hyphenation is used according to the dictionary or a style guide but must be consistent throughout the text.

Overuse of certain words is a common thesis fault that can be reduced by using a dictionary and a word count program. For example, the word **note** or phrase **note that** tend to be used too much. Other pet combinations known to irritate readers are **because** and **since**, essentially the same in mathematics, but not so for persons trained in other disciplines. A thesaurus can be useful for finding other combinations of words that replace the troublesome words.

A spelling program like `ispell` on the Unix systems is a safeguard against simple typographical errors. Other spell programs exist on microcomputer hardware. Most are useful, especially WordPerfect and MS-Word spelling checkers.

The most commonly misused controls are

```
\eject
\newpage
\clearpage
\vspace
\vskip
\hskip
\hspace
\\
\\[2pt]
\mbox{...}
\smallskip
\medskip
\bigskip
```

As a general rule, leave hard space controls out of the document! Use environments and control sequences wherever possible. The `newpage` command creates a document that the thesis office will have to reject: the page heights are to be identical.

Use `\clearpage` only to cause a dump of a page of floats (figures or tables entered with option `[p]`) or to force a figure dump at an important point. This command is a page-filling command, but it can backfire and cause figures and tables to end up on the wrong page.

Avoid using font size `\tiny`. It uses 7pt font size, which is right at the reproduction limit of 2mm for microfilm. Small subscripts in math mode can be increased in size by the use of `\displaystyle`. Commands like

```
\def\dd{\displaystyle}
```

as often seen in the preamble to decrease the typing of math mode equations, especially in the `array` environment, where compressions always seem to appear.

Error messages. There is a requirement that all section and chapter headings must be less than 4.5

inches long. This requirement has been met for the `uuthesis` style, technically, but you can enter a longer title, and hence explicit line breaks are required to make the title into an inverted pyramid.

The errors in this case show up in the preview but also in error messages during the TeX compile, especially when the TOC is input. Please *don't edit the TOC*! It is recreated every time TeX runs. The place to edit is the sectioning commands deep inside the LaTeX document, where probably no error message is emitted. The fix is to insert an optional argument `[...]`, for example,

```
\section{Main results on normal subgroups}
```

becomes

```
\section[Main results on reduction \protect\\ of normal subgroups]
        {Main results on reduction of normal subgroups}
```

It takes two runs of LaTeX to put the new TOC into the DVI output. On the second run the error given during read of the TOC should vanish.

The control `\def\mainheadingwidth{4.25in}` is standard and you may have to define it as 4.5in or 4.8in to make a particular title into an inverted pyramid. After such exceptions, it is important to set it back to 4.25in prior to typesetting the next title.

The entry in the table of contents may have to be crafted independently of the actual title in order to create line breaks that are acceptable. This is done by the optional argument to a sectioning command:

```
\chapter[Isomorphisms and Polymorphisms Revisited:
         \protect\\A Survey]
        {Isomorphisms and Polymorphisms
         \protect\\ Revisited: A survey}
```

The sectioning commands that take the optional argument in square brackets are `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`. In addition, most commands that automatically enter data into the table of contents also take a square bracket optional argument, for example, `caption` in the table and figure environments.

The coding of chapter headings followed immediately by a section head is controlled by control

`\fixchapterheading`.

If you did not read these remarks above, then do so now.

Captions on figures and tables have been revised several times. Long captions are centered, and the `\protect\\` command can be used in both the TOC and TITLE arguments. Long captions are completely feasible, but explicit line breaks may be required for extra long titles. There are two styles for captions: paragraph and centered.

Default figure/table placement can be wrong. LaTeX tries to put the figure/table on the same page following the reference with option `[b]`. Use of the `[t]` option after previewing is usually sufficient to move the figure to the next page, if it didn't fit on the right page to begin with. It helps to have the figure follow the reference as close as possible (i.e., within the paragraph, right after the sentence with the reference). Sometimes `\clearpage` is needed to force a figure onto the right page, especially at the end of a chapter.

It is required by the thesis office that a figure appear **after** it is cited, and never before the citation, even at the top of the present page. So sometimes option `[t]` doesn't work and it has to be changed to `[b]` or `[p]` or `[h]`.

The `figure*` and `table*` environments have been removed, since the format does not support double column printing in the first place.

Double and other spacing in tables may be gotten using `\arraystretch`. Default is for single spacing. Table format does require at least one horizontal line between caption and body; use `\hline`.

This style has been used with a number of different laser printers, including Imagens, Laserwriters, and Laserjets. The different engines used by these printers may cause the margins to differ when printed out on two different printers by as much as 2-3 character widths. This may be corrected by issuing print commands with offsets (if that is possible), or by overriding the default margins.

The signature pages required for theses and dissertations are formatted directly here, although you can get them from the thesis editor. These forms are correct now, but if the format of the signature pages changes, this style will have to be modified, or else you'll have to get the forms from the editor. Correspondence on changes is most appreciated (see below for contacts).

13 Bug fixes and improvements

The initial style files and documents were produced by Stan Shebs in 1987. Several improvements were incorporated with bug fixes until version 1.2 in 1988, when the style files stabilized. Since that time, only minor fixes have been added. The most recent, by Nelson Beebe, 1990-1993, and Conner and Gustafson, 1992. A major revision took place in October, 1993 by Gustafson. The fixes:

Rewrote APPROVAL pages to match the Handbook.
Rewrote 11pt and 12pt options, eliminated uut11.sty, uut12.sty.
Added `topics` environment.
Made special font sizes for all sectioning and header commands.
Built commands for singlespaced and doublespaced titles.
Rebuilt threelevels, fourlevels, fivelevels.
Rebuilt parameters for sectioning commands to match thesis manual.
Rebuilt `chapter` and `part` commands to correct TOC entries.
Fixed maintext command to be useful everywhere (standardized).
Fixed Part command so page numbering is correct.
Fixed figure problems with positioning.
Fixed figure captions and added paragraph/centering options.
Isolated theorem control sequences to make them selectable.
Added control sequences for chapter heading width.
Changed TOC entry indent to 0in. Fixed TOC entry widths.
Appendix controls rewritten. Added `noappendix` control.
Rewrote documentation, internal and external.
Add double spacing for multi-line chapter titles (UofU thesis office request 1992).

Fixed REFERENCES spacing, ABSTRACT spacing.
Paragraph style parameters now have glue removed.
Proof environment added. Theorem environments added.
Titlepage repairs.
Appendix repairs.

If you find something broken in the style file, or find a severe limitation, then please send email to one of the names below, so that the changes and fixes get applied to the master files.

14 Author addresses

If you have problems with this format, or suggestions for improvements and features, contact (in this order):

Nelson Beebe	Math, 581–5254	beebe@math.utah.edu, 1993
Grant B. Gustafson	Math, 581–6879	gustafson@math.utah.edu, 1993
Leigh B. Stoller	CS, 585-3733	stoller@cs.utah.edu, 1993
Stan Shebs	Apple (CA)	shebs@cygnus.com, 1993

15 Acknowledgments

The uuthesis style file has passed through many hands. It seems to have started life both as the standard LaTeX report style file report.sty by Leslie Lamport, and as a Stanford thesis format suthesis.sty, done by an unknown person at Stanford. Mike Jordan adapted it to the University of Utah style and wrote the format of signature pages. Mark Watkins did some cleanup, and added macro definitions to parameterize committee members, graduate deans, and so forth, as well as writing a few of the paragraphs above. Other contributions have been made by Nelson Beebe, Ed Cetron, Greg Conner, David Eyre, Grant Gustafson, Ming He, Paul Joyce, Tina Ma, Jeff McGough, Eric Muehle, John W. Peterson, Robert Dillon and Jeff Yost. Finally, Linda Zillman, thesis editor, was very helpful in adapting the format to the capabilities and restrictions of LaTeX. Continued support in 1993 has kindly been given by Christine Pickett, thesis editor, and many of the newest features and fixes are due to her enthusiasm and cooperation.

16 Sample Root Files

The following is a root file for a dissertation. Made 1993 by Jeff McGough and Grant Gustafson. It uses separate files for everything, including the abstract, acknowledgments, chapters and bibliography. Note in particular the preamble items versus the control sequences that appear after the start of the document.

```
\documentstyle[11pt,amssymbols,thesis,diagram,tgrind]{uuthesis2}
\includeonly{chap1,chap2,chap3,appendix}
```

```
\title{REPRESENTATION OF SOLUTIONS TO LINEAR\protect\\
       ALGEBRAIC EQUATIONS}
\author{Fred Krylov}
\thesistype{dissertation}
\graduatedean{Ann W. Hart}
\department{Department of Mathematics}
\degree{Doctor of Philosophy}
\departmentchair{Paul Fife}
\committeechair{Fletcher Gross}
\firstreader{Hans Othmer}
\secondreader{Jim Carlson}
\thirdreader{Grant Gustafson}
\fourthreader{Nick Korevaar}
\chairtitle{Professor}
\submitdate{March 1993}
\copyrightyear{1993}
\fourlevels
\dedication{For my cat, Mouse, a few lines only.}
 \inputpicturetrue  % By Jeff McGough.
\begin{document}
\frontmatterformat
\titlepage
\copyrightpage
\committeeapproval
\readingapproval
\preface{abstract}{Abstract}
\dedicationpage
\tableofcontents
\listoffigures
\listoftables
\optionalfront{Notation and Symbols}{\input{notation.tex}}
\preface{acknowledge}{Acknowledgements}
\maintext
\include{chap1}
\include{chap2}
\include{chap3}
\numberofappendices=1
\appendix
\include{appendix}
\bibliographystyle{siam}
\bibliography{thesis}
\end{document}
```

Eyre Thesis. The following is the contents of `thesis.tex` for the Masters Thesis of David Eyre, March, 1990. It is a numerical analysis thesis with a large section of computer code. David produced the thesis on a SUN workstation and also on the VAX 8600 mainframe computer. It contains no figures. The thesis was first typed in ARTICLE style as several separate documents and then re–edited into the thesis format. It is an unusual example in that it contains no list of figures, no list of tables and only a single appendix.

In the actual thesis produced for the library, the computer utility `lptops` was used to generate the thesis pages for the computer code, complete with page numbering. The file `code.tex` simply inserts the correct page number so that the bibliography is numbered properly.

```
\documentstyle[12pt]{uuthesis}
\thesistype{thesis}
\title{EXACT ANALYSIS OF GENERAL SPARSE \\
~ \\ RATIONAL LINEAR SYSTEMS}
\author{David Jay Eyre}
\degree{Master of Science}
\department{Department of Mathematics}
\graduatedean{B. Gale Dick}
\submitdate{March 1990}
\copyrightyear{1990}
\committeechair{Peter Alfeld}
\firstreader{Paul C. Fife}
\secondreader{Don H. Tucker}
\departmentchair{Klaus Schmitt}
\dedication{\mbox{For Cas}}
\chairtitle{Professor}
\threelevels
\newtheorem{th}{Theorem}
\newtheorem{lm}[th]{Lemma}
\newtheorem{co}[th]{Corollary}
\newtheorem{df}[th]{Definition}
\begin{document}
\requiredfrontmatter{abstract}
\preface{acknowledge}{ACKNOWLEDGMENTS}
\maintext
\input{tbody}
\input{appendix}
\input{code}
\input{biblio}
\end{document}
```

Joyce Thesis. The following is the contents of `thesis.tex` for the Ph.D. thesis of Paul Joyce, June, 1988. It is a mathematics thesis that does not use automatic features of the `uuthesis` style. This particular thesis was typeset using TEXtures on a MacIntosh SE computer with output from a Laserwriter printer.

Figures were automatically integrated at print time (no paste-up). Note in particular the ordering of control sequences and the placement of definitions in the preamble.

The Joyce thesis, done on a MacIntosh computer using Textures, is a large thesis consisting of the following computer source files:

File Name	Description
THESIS.TEX	main control file
ABSTRACT.TEX	abstract
ACK.TEX	Acknowledgments
THESIS.GRAPHICS	figures
CH1.TEX	chapter 1
CH2.TEX	chapter 2
CH3.TEX	chapter 3
CH4B.TEX	chapter 4
CH5.TEX	chapter 5
CH6.TEX	chapter 6
CH7.TEX	chapter 7
CH8.TEX	chapter 8
CH9.TEX	chapter 9
THESIS.BIB	bibliography

The file THESIS.TEX is reproduced below, so that you can see how to structure the writing of the thesis. Paul wrote the key chapters first, then the figures, bibliography and introduction.

```
\documentstyle{uuthesis}
\includeonly{ch7,ch8,ch9} % Debug only chapter 7, 8, 9
\title{Age--Ordered Distributions For Population Genetics
Models}
\author{Paul Joseph Joyce}
\thesistype{dissertation}
\graduatedean{B. Gale Dick}
\department{Department of Mathematics}
\degree{Doctor of Philosophy}
\departmentchair{T. Benny Rushing}
\committeechair{Simon Tavar\'e}
\firstreader{Robert M. Brooks}
\secondreader{Stewart N. Ethier}
\thirdreader{James P. Keener}
\fourthreader{Klaus Schmitt}
\chairtitle{Professor}
\submitdate{June 1988}
\copyrightyear{1988}
\dedication{This thesis is dedicated to my parents
```

```
Dorothy A. and Thomas F. Joyce.}
\threelevels
\newtheorem{lemma}[theorem]{Lemma}
\begin{document}
\frontmatterformat
\titlepage
\copyrightpage
\committeeapproval
\readingapproval
\preface{abstract}{ABSTRACT}
\dedicationpage
\tableofcontents
\listoffigures
\preface{ack}{ACKNOWLEDGMENTS}
\maintext
\include{ch1}
\include{ch2}
\include{ch3}
\include{ch4b}
\include{ch5}
\include{ch6}
\include{ch7}
\include{ch8}
\include{ch9}
\bibliographystyle{acm}
\end{document}
```