USING LaTeX TO PRODUCE MASTER'S THESES AND DOCTORAL DISSERTATIONS IN ELECTRICAL AND COMPUTER ENGINEERING

BY

ECE PUBLICATIONS OFFICE

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Documentation Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Doctoral Committee:

Editor James A. Hutchinson, Chair
Editor Jerome S. Colburn
Professor Peter J. Lea
Professor Carol D. Matte

ABSTRACT

LaTeX is a powerful system for producing electronically formatted print matter given a plain text input file with embedded commands introduced by special characters. Particularly when mathematical content is present, LaTeX has no match in output quality among ordinary desktop word processors. Therefore it is favored and recommended for master's theses and doctoral dissertations as well as journal articles and books.

To use LaTeX's power, however, the student must learn somewhat of an actual programming language. The present thesis is intended as an introduction to LaTeX and how to use it to produce a master's thesis or doctoral dissertation in the Department of Electrical and Computer Engineering (ECE) at the University of Illinois at Urbana-Champaign using the specialized document class `uiucecethesis09`.

To you, the student

ACKNOWLEDGMENTS

Thanks to Donald E. Knuth, Leslie Lamport, and many others who produced the great body of TEX software and released it to the public free of charge; to David Hull, Peter Czoschke, and others in the Physics Department who developed the document class `uiucthesis`, from which `uiucecethesis09` was derived; to Shane Ryoo, who first made the ECE modifications; to Jerome Colburn for creating the first "official" ECE Department LATEX files and for drafting this thesis; to J.P. Peters for taking care of the thesis writers; and to Matt Carroll for the printing.

TABLE OF CONTENTS

Note: As of 3 October 2011, the Graduate College will no longer accept Author's Biography, CV, or Vita in a thesis or dissertation.

CHAPTER 1

LaTeX: WHAT AND WHY

This chapter describes the origin and *raison d'être* of TeX and LaTeX, followed by a few words about its user interface, output formats, and fonts. Chapter 2 describes the basic structure of any LaTeX document, the concept of a LaTeX "template," and the template for theses in ECE. Chapters 3 through 6 describe in particular how to put together a thesis or dissertation in ECE using LaTeX, complete with figures, tables, bibliography, and title page in the format required by the Graduate College.

1.1 Why TeX?

About thirty years ago, Donald Knuth, professor of computer science at Stanford University, was dissatisfied with the quality of the typesetting on his six-volume treatise *The Art of Computer Programming* and resolved to write his own software to do a better job of it. The result was, and is, TeX (in which the last letter is a capital Greek letter chi, hence the pronunciation "tech"). Given a plain text file as input, in which formatting instructions are marked using special characters—particularly `\`, `{`, `}`, and `$`—TeX automatically fills out lines, paragraphs, and pages; hyphenates automatically as well as possible; and sets fonts and character styles.

From the outset TeX has been optimized to handle complicated math:

$$\nabla \bullet \mathbf{D} = \rho \tag{1.1}$$

$$\nabla \bullet \mathbf{B} = 0 \tag{1.2}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{1.3}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \tag{1.4}$$

...and there was light. This optimization is one reason for the continuing popularity of TEX (and LATEX) among engineering and physical science students in the face of competition from such products as Microsoft Word. Another reason is that TEX and LATEX are free. Related to this reason is the fact that Knuth and the TEX user community are not interested in coming out with a new version every few years that entices users to create files that earlier versions of the program cannot handle and therefore force the rest of the users to buy the new version. A third reason is that TEX and LATEX are boundlessly extensible by means of user-defined macros. Many useful macros are already part of the TEX package (including the macro `\TeX` itself that produces the special logo "TEX"). A handy guide to TEX is [1].[1]

1.2 Why LATEX?

TEX, however, does not automatically support logical document structure. Plain TEX can be used to write a document with various levels of sections with their headings in a brute force manner: when users come to write a section heading, which they've decided should be in, say, 14-point bold font with two blank lines' worth of space above and below, they hard-code the commands to set the font size to 14pt and the style to bold and to set the requisite blank space. This approach is similar to what many users of Microsoft Word do, setting the font size and style and the paragraph spacing using the conveniently provided toolbar and ribbon buttons. The TEX users *could* write macros to set these styles, just as Word users *could* (but most do not) use Word's style definition features to create the styles they want for the elements they need.

LATEX (pronounced "lah-tech" or "lay-tech"), invented by Leslie Lamport, is, essentially, a collection of macros that sits on top of the TEX engine to solve these problems. It provides intuitive commands for creating document structure. To start this chapter, for example, all I had to do was type

[1] Knuth's own documentation [2] attempts to combine a tutorial with a complete reference manual. It does not altogether succeed in doing so; the learner will see a great many features that relate to more advanced topics, while the experienced user who needs information on a particular topic must delve deep into the (fortunately quite extensive) index, often to find the necessary point covered almost in passing in a small-type paragraph marked with a highway "dangerous curve" symbol in the middle of the discussion of some other feature.

`\chapter{\LaTeX: What and Why}` and hit the Enter key. When I ran LaTeX on the file, it automatically knew to start a new page, set extra space at the top of the page, set the correct chapter number in sequence, set the chapter number and title in large capitals, set more space below the chapter title, add a line for the chapter to the table of contents, and not indent the first paragraph. It also knew to set the correct chapter number in the section numbers of the section heads in this chapter (created by typing `\section{Why \TeX?}` and `\section{Why \LaTeX?}`, respectively) and in the equation numbers of Equations (1.1)–(1.4) in Section 1.1.

Those equation and section number citations in the last sentence of the preceding paragraph were also produced (semi-)automatically. I put the command `\label{sec-whytex}` after the `\section` command for Section 1.1 and the commands `\label{eq:maxdivE}`, `\label{eq:maxdivH}`, `\label{eq:maxcurlE}`, and `\label{eq:maxcurlH}` on the respective lines for the four Maxwell equations (1.1)–(1.4). Then, to cite the section and equations, I typed `Equations (\ref{eq:maxdivE})--(\ref{eq:maxcurlH}) in Section \ref{sec-whytex}`. (For more on LaTeX's cross-referencing capabilities, see Section 2.5.2.)

TeX (and therefore LaTeX) was designed from the outset to *justify* the type on each line automatically: that is, to make the left and right edges of a paragraph line up by breaking and hyphenating words and inserting extra space between words to fill out each line. In addition, LaTeX also automatically indents each paragraph except after a heading, as in well-produced books. (Special effort is required to defeat these features, but it can be done if necessary.)

As mentioned earlier, LaTeX automatically produces a table of contents, which the user can place at the front of a book or report, ordinarily showing all chapter numbers and titles, section numbers and titles, and subsection numbers and titles. LaTeX also provides features for inserting tables and figures that can be automatically numbered in sequence within each chapter and for automatically creating lists of tables and figures that can be placed at the front of a book.

1.3 Operating LaTeX

1.3.1 User interfaces

LaTeX works by means of commands embedded in text files that are processed by particular programs that were originally invoked from command line user interfaces, particularly the shells in the various UNIX flavors. This type of process can be bewildering to a user accustomed to pointing and clicking in a graphical user interface (GUI) such as Microsoft Windows.

Fortunately there are now GUI integrated development environments (IDEs) for TeX and LaTeX, similar to those used by C++ and Java programmers. This thesis was produced using the TeXnicCenter IDE for Windows, which is supplied with the ProTeXt distribution available from [3]. The user interface is shown in Figure 1.1. At upper right is the editing window, which highlights TeX and LaTeX commands in color. The document structure in chapters and sections is conveniently displayed as a tree at left. At a click of a button, or a press of Ctrl+Shift+F5, the file currently open is saved; TeX and other necessary programs are run on it; the TeX log, showing the progress of processing and any errors, is displayed in the bottom window; and the finished product (such as a PDF file) is displayed in the proper viewer (such as Adobe Reader).

The toolbars at the top of the window contain many useful buttons. In addition to the buttons usually found on any GUI (e.g., save, open, new document), there are buttons to generate and view the formatted output, find errors, and insert code automatically for lists, tables, character styles, paragraph alignments, and math symbols.

1.3.2 Output formats

By itself, TeX (and therefore LaTeX) outputs a file containing only very precise placement instructions for characters and graphics. It cannot be sent directly to any particular printer, and therefore it is said to be "device independent" (DVI), which is why it has the extension `.dvi`. The various TeX distributions have DVI driver utilities, which make it possible to view such files on the screen or print them, but the resolution is limited.

A more useful option for handling the DVI files is to convert them to

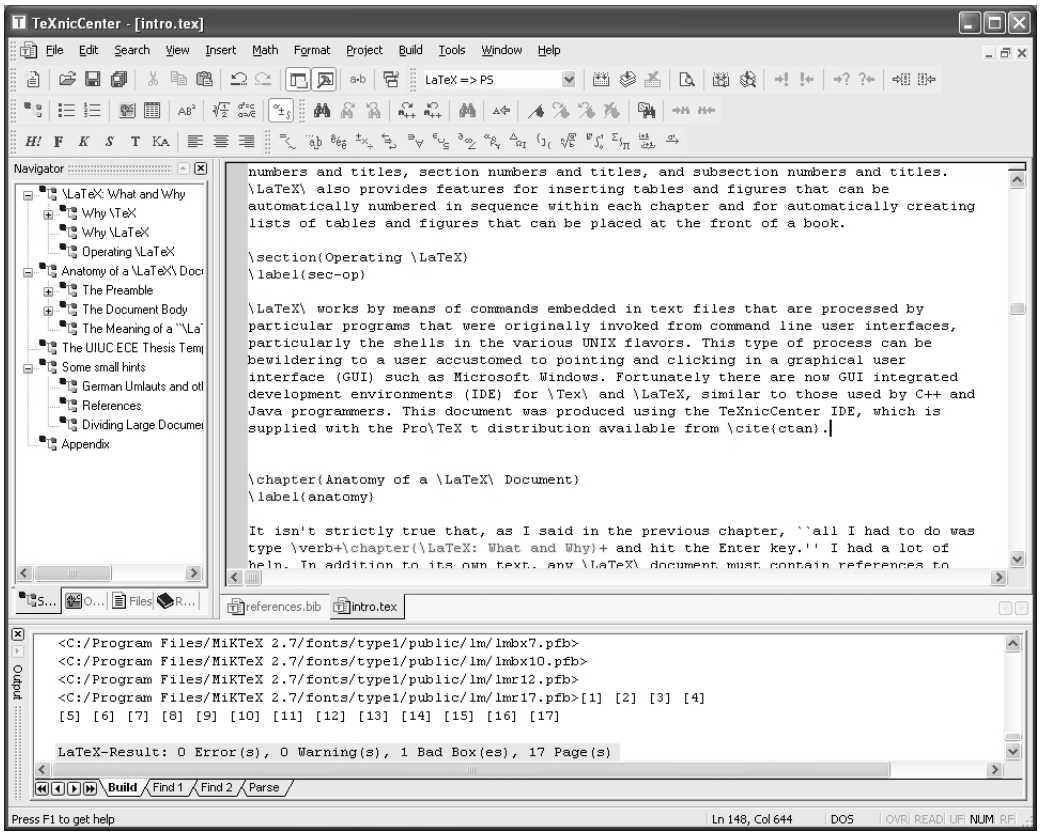


Figure 1.1: The TEXnicCenter user interface.

Adobe PostScript (PS) using the utility `dvips`. PostScript files can be sent to a PostScript printer, or they can be displayed on the screen, printed, or converted to Adobe Portable Document Format (PDF) using a program such as the free Ghostscript and GSview utilities.

If the TEX output is converted to PS, the fonts must exist in Adobe Type 1 form. Type 1 versions of all the fonts commonly used on TEX projects are provided with most TEX distributions.

The ProTEXt distribution includes Ghostscript, GSview, and the Type 1 fonts. The installation process connects them all up with the TEXnicCenter IDE so that one can obtain PS and PDF output at will. (If Adobe's Acrobat Distiller is available on the system, TEXnicCenter will use it to produce PDF.) For example, simply setting the output profile to `LaTeX=>PS` using a drop-down menu at the top of the window and pressing Ctrl+Shift+F5 runs TEX and pops up the result in GSview (unless an error prevents output from being generated).

If you have a PS file open on display in GSview, then use another program to make a change in the TEX source file, regenerate the PS, and then switch back to the GSview window to see the results, GSview will automatically reload and redisplay the changed PS file. By contrast, if you are producing PDF to view in Acrobat, you must close the previous version of the PDF file in Acrobat before rerunning TEX; otherwise, the PDF will not be regenerated. For this reason, the `LaTeX=>PS` and `LaTeX=>DVI` output profiles (especially the latter) are preferable while the document is being written and edited. After editing is complete, switch to the `LaTeX=>PS=>PDF` profile and get the PDF.

1.3.3 Fonts

Along with TEX, Knuth created a program called METAFONT [4], which takes a plain text source file containing point-by-point descriptions of the outlines of characters and produces files containing bitmap images of the characters. The intent was that the DVI driver, given a DVI file and a printer specification, would automatically invoke METAFONT to produce the character bitmaps to be placed in the final image. Nowadays it is almost always more convenient to use PostScript, and there are Type 1 versions of

the fonts most commonly used with TEX and many more besides, so METAFONT is rather antiquated. If a TEX document using a font that does not have a Type 1 version is used to produce a PS file, METAFONT will still be used to produce bitmap images, but they will show up as jagged and distorted depending on the resolution of the PS display, in contrast to the smooth outlines of the PS fonts.

Knuth also wrote the METAFONT source files for the Computer Modern (CM) fonts used in his textbook. They are the default fonts that will be used if the TEX or LATEX user does not do anything to specify fonts. They are used in [1], [2], [5], and the present document. They exist in Type 1 forms for use in producing PS output.

The CM roman and *italic* fonts in the body of this document admittedly do not look very "modern." They are patterned after fonts typical of math books from the first half of the twentieth century, and some find their look rather stuffy. Source files used by TEX and LATEX can be tweaked to substitute Times New Roman or other fonts to get a more "modern" look, but *this is not recommended for documents that contain math.* If only the fonts used for text are set to the new font and those used for math are left untouched, the output will be inconsistent—the same character (such as x) will look different depending on whether it was input as italicized text (`\textit{x}`) or as math (`$x$`). Math symbol fonts that match the appearance of the new font may be available only at extra cost or not at all, while the math fonts corresponding to the look and feel of CM are freely available as part of most TEX distributions.

CHAPTER 2

BASIC ANATOMY OF A LaTeX DOCUMENT

It is not strictly true that, as stated in the previous chapter, "all I had to do was type `\chapter{\LaTeX: What and Why}` and hit the Enter key." I had a lot of help. In addition to its own text, any LaTeX document must contain references to several other files that collectively tell LaTeX how to handle the various elements.

2.1 LaTeX Text and Commands

A LaTeX document consists of characters that either are plain text characters or are to be read as part of a command.

2.1.1 Text

Plain English text is, in general, simply typed into the document file as is. There are a few exceptions.

- To use opening and closing curly quotation marks "like this" rather than two closing ones "like this" (a common error among LaTeX novices), type ` `` ` for the left quote and `''` for the right quote. The ` ` character is on the key at the upper left corner of a standard keyboard, which also has the `~` symbol. The `'` character is the ordinary apostrophe.
- To set an en dash (intermediate dash), as in "the years 2009–2013," type two hyphens `--`.
- The minus sign (−) does not look the same as a hyphen (-) and does not necessarily look the same as an en dash (–). Setting a minus sign is discussed in Chapter 3.

- To set an em dash (punctuation dash)—used, as in this sentence, to set off an interrupting comment—type three hyphens `---`. In the Computer Modern fonts (Section 1.3.3), the em dash has the property that adjacent em dashes form a solid line: ——— was produced by typing `{---}{---}{---}`.

- To prevent a space from breaking at the end of a line (for example, between initials), type a tilde `~` instead of a space: `E.~B.~White` to get "E. B. White" without a line break.

- Multiple successive spaces are printed as one space.

Other exceptions are addressed in the following sections.

One very important special character is the percent sign `%`. If this character is encountered in a TEX or LATEX document, TEX will ignore the rest of the line, treating it as a *comment* rather than text to be typeset. This feature has the same uses as do comments in programming languages such as C++ or Java:

```
% This is a comment
```

After the end of a paragraph, type a blank line before going on to type the next paragraph text. Just hitting Enter once, as would be done in Word, will not be enough; without the blank line LATEX runs the two consecutive "paragraphs" into one.

2.1.2 Commands

A vast repertory of commands is available in LATEX, some of which are built into the basic TEX engine, others are built into the LATEX software, and yet others are defined in the document style file and packages used by a particular document.

The mark of a command is the backslash `\`. After the backslash, the command itself consists of either

- One or more letters, up to the first following character that is not a letter (e.g., `\TeX`)

- A single character that is not a letter (e.g., `\{`)

If a command consisting of one or more letters is followed by one or more spaces, the spaces terminate the command word but are not printed in the text. Thus, `\TeX as a state of mind?` will print as "TEXas a state of mind?" To force a space, use the special command consisting of a backslash followed by a space, `\ : \TeX\ as a state of mind?` produces, correctly, "TEX as a state of mind?"

After the command word may come an *argument,* a piece of information that the command needs to do its job. The argument to the command is included between two curly braces `{}`. For example, as was seen in the preceding chapter, the `\chapter` command requires the chapter title: `\chapter{Basic Anatomy of a \LaTeX\ Document}`. The argument to a command can be a single word or can extend over many paragraphs, and it can itself contain any number of commands with their arguments.

Probably the most common error made in entering a TEX or LATEX document is to leave out one of the curly braces `{` and `}`. This error confuses TEX mightily and can lead to many pages of incorrect output or no output at all. A large number of error messages, all consequences of a single omission, may result.

Curly braces can also be put around material other than the argument of a command. Doing so sets the enclosed material off so that it is processed all together and terminated without disturbing the surrounding text. For example, the phrase "TEX as a state of mind?" could be obtained by putting the command `\TeX` inside curly braces: `{\TeX} as a state of mind?` The closing curly brace terminates the command, and the following space outside the braces is preserved.

If it is necessary to use a curly brace character itself in the text, put a backslash before it: `\{you\}` produces {you}. This is the usual way to print special characters that TEX and LATEX use as something other than printing characters. A pitfall is the backslash itself. `\\` does not produce a backslash character, as one might expect, but a forced line break. To get a backslash character, it is necessary to go into math mode (Chapter 3) and use the `\backslash` command: `$\backslash$` produces \.

Other commands consisting of characters other than letters include those traditionally used for producing diacritical marks, such as `\'` for an acute accent, useful in writing your résumé.

2.1.3 Environments

Commands are used by both plain TEX and LATEX. A feature unique to LATEX is the *environment.* An environment has its own special formatting rules and may have special commands that work only within the environment or redefine standard commands to work in a special way inside the environment. To enter an environment, use the command `\begin` with the name of the environment as the argument. To exit the same environment, use the command `\end` with the name of the environment as the argument. For example, bulleted lists such as the list of differences between document classes and packages in Section 2.2.2 are created using the `itemize` environment. The command `\begin{itemize}` tells LATEX to set the extra space at the top of the list and to define the command `\item` to mean one list item and set the leading bullet automatically. At the end of the list, `\end{itemize}` tells LATEX to set the extra space at the bottom and forget about the definition of `\item`.

Environments can be nested within environments. For example, all the text, commands, and environments used in this thesis are enclosed in one large `document` environment.

In addition to the environment name, a `\begin` command can have a second required argument after the environment name and can have an optional argument (or argument list) set in square brackets *before* the environment name. The presence or absence of these additional arguments depends on the particular environment.

2.2 The Preamble

The *preamble* of a LATEX document consists of everything in the file up to the beginning of the `document` environment (the `\begin{document}` command). The preamble contains commands that bring in definition files (the document class and packages), define or redefine commands and environments, or set constant values to be used later in the document, but it contains no text or other content elements.

2.2.1 The document class

The first line in the document is a `\documentclass` command. This command ends with a word enclosed in curly braces `{}`. That word is the file name of a *document class* definition file, to which the extension `.cls` is appended. Thus, the `\documentclass{report}` tells LaTeX to look up the file `report.cls`, which comes standard with the LaTeX installation. The document class file contains the macros that define how such things as chapter openers, tables of contents, and title pages will be displayed in this document.

Between the word `\documentclass` and the braces containing the class file name, the `\documentclass` command ordinarily has a list of *options,* enclosed in square brackets `[]` and separated by commas. The available options are defined in the document class file. For example, this document begins with

```
\documentclass[largecaps,tocnosub,noragright,12pt]{uiucecethesis09}
```

See Section 2.4 for specific information on these and other options.

A standard LaTeX distribution contains the classes `article`, `book`, `letter`, `report`, and `slides`. The `uiucecethesis09` class, for example, ultimately derives from the `book` class.

2.2.2 Packages

After the `\documentclass` command, the preamble can and usually does contain one or more `\usepackage` commands, which invoke *package* definition files rather than document class definition files. Just to be confusing, a package definition file has the extension `.sty` (short for "style"). Like the `\documentclass` command, the `\usepackage` command ends with the file name in curly braces, which may be preceded by a pair of square brackets containing a list of options defined in the package file. For example, the first line after the `\documentclass` command in this document is

```
\usepackage{graphicx}
```

This command invokes the file `graphicx.sty`, which enables inclusion of external graphics files.

The differences between a document class and a package are the following:

- A LaTeX document can have only one document class but may use any number of packages.
- A document class applies to only one kind of document, whereas a package provides some functionality that can be used in documents belonging to any number of classes.

There are very many packages in a standard LaTeX distribution. Many of these are described in [6]. Many more can be found online.

2.2.3 Macro definitions and redefinitions

LaTeX commands and environments are known as *macros* because they map a single command to a sequence of multiple other commands and text. The `\documentclass` and `\usepackage` instructions in the preamble import files containing macro definitions. It is also possible to put macro definitions or redefinitions in the preamble, and even in the document body as well. The commands `\renewcommand` and `\renewenvironment` are used to redefine commands that have already been defined in the document class file or one of the packages used; for example, to redefine the chapter opener command to use a different font for the chapter title. The commands `\newcommand` and `\newenvironment` are used to define new macros that are not defined in the document class file or any of the packages. Given a correctly designed document class and well-chosen packages, the user should not need to do any such defining or redefining. Anyone attempting to write such definitions should be familiar with [5] and [6], and [1] will be useful as well.

2.3 The Meaning of a "LaTeX Template"

In Microsoft Word, a "template" is a single file (`.dot` in Word 2003 and earlier, `.dotx` or `.dotm` in Word 2007 and later) that contains style specifications (e.g., paragraph style Normal might mean 12 point Times New Roman, single spaced, with justified margins, no indentation, no extra space above or

below the paragraph) and perhaps macro definitions and toolbar customizations. It also contains the overall document layout. Double-clicking on the template file icon in a folder window opens a new document that complies with the template and offers all the features programmed into the macros and styles. (The document is said to have the template "attached.")

Accomplishing the same purposes in LaTeX is more complicated. It requires at least

- A document class (`.cls`) file
- A blank starter document (usually `.tex` or `.ltx`) file, which contains at least the commands needed for the skeletal structure of a minimal document and should also contain copious comments mentioning the various options, commands, and environments offered by the document class

There may be one or more additional package (`.sty`) files. For bibliographies there may be custom BibTeX style (`.bst`) files and starter bibliography (`.bib`) files; see Chapter 6 for explanations of these files.

If the document class, packages, and BibTeX styles are properly written, it should *never* be necessary for the user to change anything in those files. Everything the user needs should be available by using suitable commands in the LaTeX document.

In addition to custom package and BibTeX files, the "template" can and almost always does rely on a large number of such files that are already provided by an up-to-date LaTeX distribution. The starter file should contain the necessary `\usepackage` commands for both custom and standard packages required by the document class.

There should also, of course, be one or more documents that describe how to use the class and package features. In fact, the ordinary form in which a LaTeX document class starts life is as a file with the extension `.dtx` that contains not only the code for the class but also complete documentation in the form of comments. The file belongs to the class `ltxdoc`. The comments, introduced by the percent sign and a space as usual, are themselves LaTeX coded and use commands and environments that format the code for display when LaTeX is run on the `.dtx` file. There is also a utility program `docstrip`, which, when run on the same `.dtx` file, produces the document class file with

all the comments stripped out. Use of a `.dtx` file encourages programmers to write their code and their documentation at the same time, with the hope that the documentation will explain the code after the programmer has ceased to work on it. However, the documentation so written tends to be primarily a reference manual for other programmers maintaining the class file themselves. At best it answers the question "What does *this command,* specific to this document class, do?"

A student trying to write a thesis, on the other hand, needs a document that answers the question "How do I perform *this task,* whether by new commands specific to this document class or in some other way?" This thesis itself is intended to be such an introduction.

As of late 2009, the "UIUC ECE thesis template" consists of the following files: the document class file `uiucecethesis09.cls`; the starter document file `ecethesis.tex`; the IEEE bibliography style file `IEEE_ECE.bst`; the starter bibliography file `thesisrefs.bib`; and this thesis.

2.3.1 Document class `uiucecethesis09`

The `uiucecethesis09` document class is based on the `uiucthesis` class, which is available from the UIUC Department of Physics in the form of a `.dtx` file as mentioned in Section 2.3.

The earlier `uiucecethesis08` class, used through the spring semester of 2009, was also based on `uiucthesis` but was not made by changing a `.dtx` file. It contained changes by Shane Ryoo and the present writer to bring the thesis format into conformity with ECE and recent Graduate College requirements. These changes have all been transferred to `uiucecethesis09.dtx`, from which the `uiucecethesis09` class is generated.

2.3.2 Starter file `ecethesis.tex`

The document starter file `ecethesis.tex` is a blank thesis that has all the elements in place to serve as a starting point for writing your thesis. Fill in your text and information, and delete or comment out the lines you do not need.

The starter file is set up to encourage you to write your chapters and

other structural elements in separate files, using the `\include` command to incorporate them in the final document, so that the main thesis file remains short and skeletal (see Section 2.10.1 to learn how to do so).

2.4 Options in the UIUC ECE Thesis Document Class

When you begin to write your thesis in LaTeX, you are encouraged to start with the file `ecethesis.tex` and fill it out. At the top of the file, in the command `\documentclass{uiucecethesis09}`, you may select from several options that primarily affect the formatting of the thesis pages.

2.4.1 Controlling the type size

The `12pt` and `11pt` options specify larger type sizes than the default 10 pt. We strongly recommended 12 pt, especially considering the fact that, after conversion to PDF, the font may well appear smaller than its specified size.

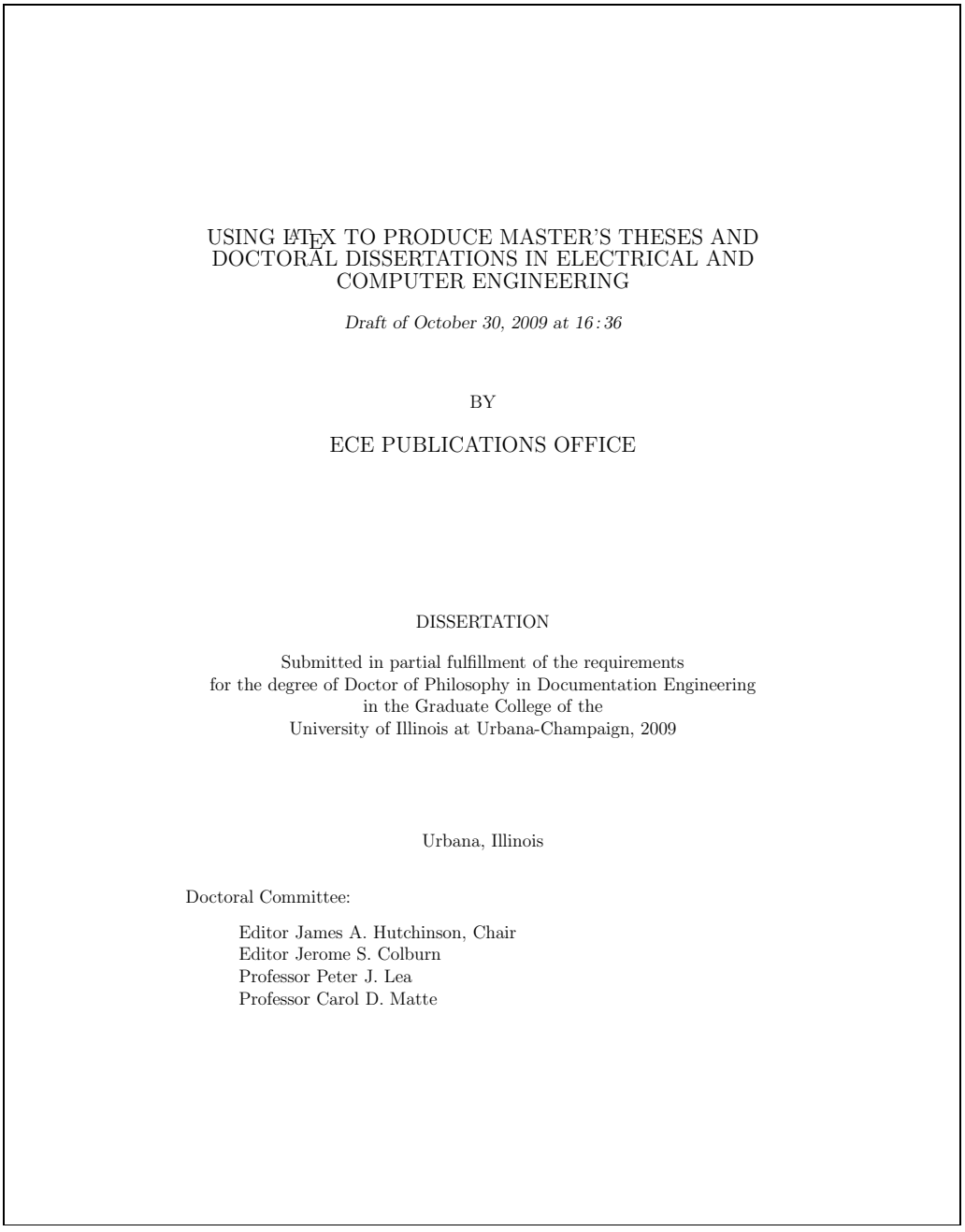
If you still wish to use a small font size and want your figure captions to appear in as small a font as the Graduate College allows, use the option `littlecaptions`.

2.4.2 Time-stamping a draft thesis

The `draftthesis` option puts a time-stamped running header on each page (including the title page) identifying the thesis as a draft, as shown in Figure 2.1. This feature is useful for distinguishing between revisions, but it must be turned off when you are ready to submit the thesis for format check.

2.4.3 Controlling margin widths

By default, the `uiucecethesis09` class produces 1.5-inch margins on both sides, which more than ensures that the Graduate College margin requirement (one inch on both sides) is met. Line spacing defaults to an intermediate between single and double. The `fullpage` option produces lines of text that extend 0.5 inch farther on either side and are double-spaced. To get these



USING LaTeX TO PRODUCE MASTER'S THESES AND DOCTORAL DISSERTATIONS IN ELECTRICAL AND COMPUTER ENGINEERING

Draft of October 30, 2009 at 16 : 36

BY

ECE PUBLICATIONS OFFICE

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Documentation Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2009

Urbana, Illinois

Doctoral Committee:

Editor James A. Hutchinson, Chair
Editor Jerome S. Colburn
Professor Peter J. Lea
Professor Carol D. Matte

Figure 2.1: Dissertation title page with the `draftthesis` option.

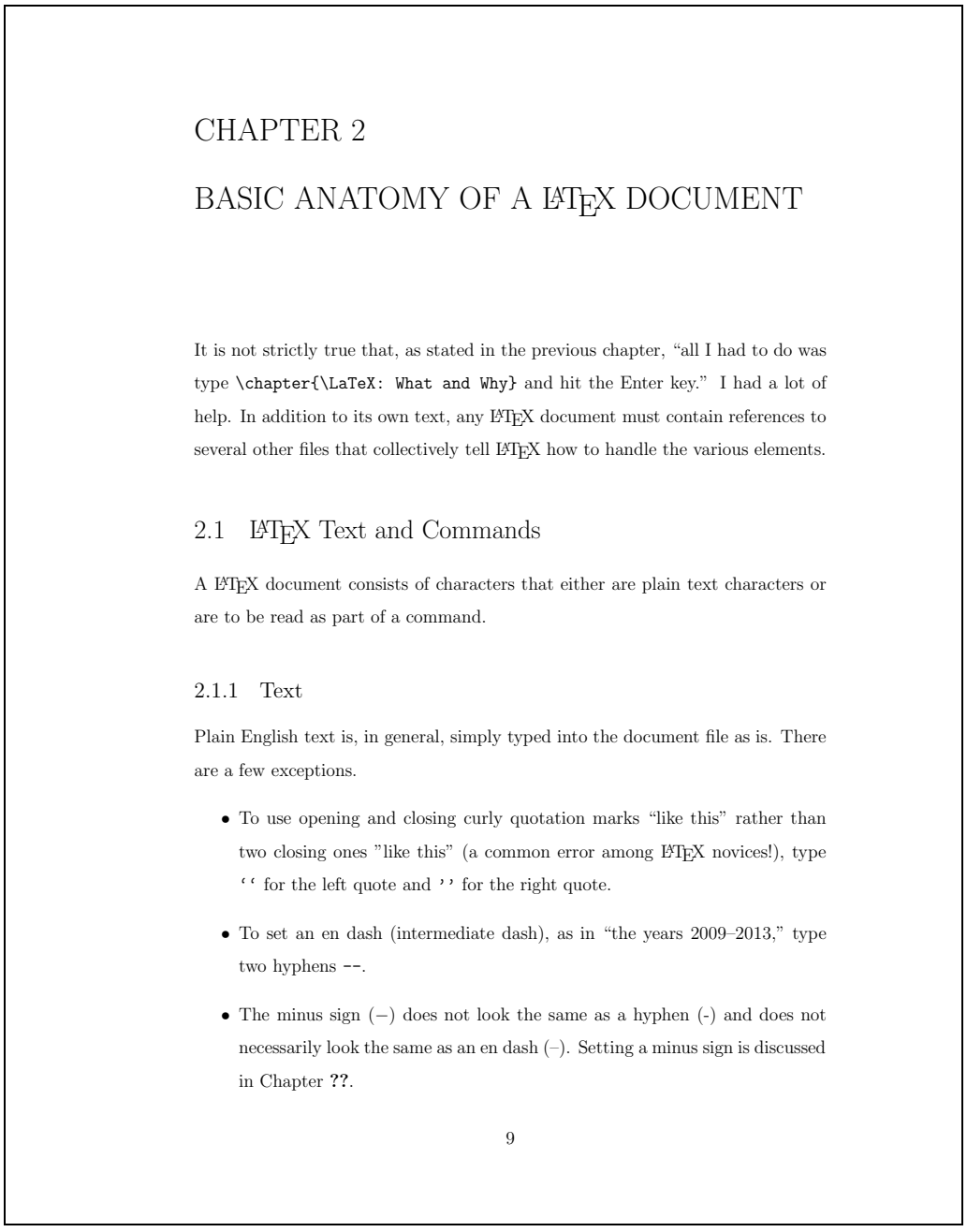
wider lines but *without* the double spacing, use `fullpagesingle` instead of `fullpage`.

Students who use the `fullpage` option and would like to keep a wide gutter for single-sided printing and binding may use the `offcenter` option to add 0.5 inch on the left side, producing the result shown in Figure 2.2. (Note that this option [Figure 2.2] is unsatisfactory for double-sided printing, in which the extra gutter space must alternate between left and right; however, for those students who still prefer to deposit a hard copy of their thesis, the Graduate College and Library require single-sided printing. Furthermore, note that the combined use of `fullpage` and `offcenter` may disrupt the appearance of the all-important title page, necessitating tricky manual adjustments.)

2.4.4 Other cosmetic adjustments

Following are other cosmetic adjustments enabled by the `uiucecethesis09` class:

- `centerchapter` produces centered titles in chapter openers.
- The `fancy` option produces chapter openers with the chapter number in a large font to the left of the chapter title as shown in Figure 2.3. This option is not recommended because the large chapter number in the margin slightly violates the one-inch margin requirement unless the `offcenter` option is selected. Also, look out for double-digit chapter numbers! (Alternatively `centerchapter` may be used, but the results are not esthetically satisfactory.)
- `toclabels` puts column headers over the Table of Contents (not recommended).
- `largecaps` sets the author and title in large font on the title page.
- `edeposit` is an old option that formatted the title page for electronic deposit (e.g., list the doctoral committee on the title page). Current Graduate College specifications require all theses to be formatted for electronic deposit (even if they are submitted as hard copy), so this option is not necessary and no longer does anything.



CHAPTER 2

BASIC ANATOMY OF A LaTeX DOCUMENT

It is not strictly true that, as stated in the previous chapter, "all I had to do was type `\chapter{\LaTeX: What and Why}` and hit the Enter key." I had a lot of help. In addition to its own text, any LaTeX document must contain references to several other files that collectively tell LaTeX how to handle the various elements.

2.1 LaTeX Text and Commands

A LaTeX document consists of characters that either are plain text characters or are to be read as part of a command.

2.1.1 Text

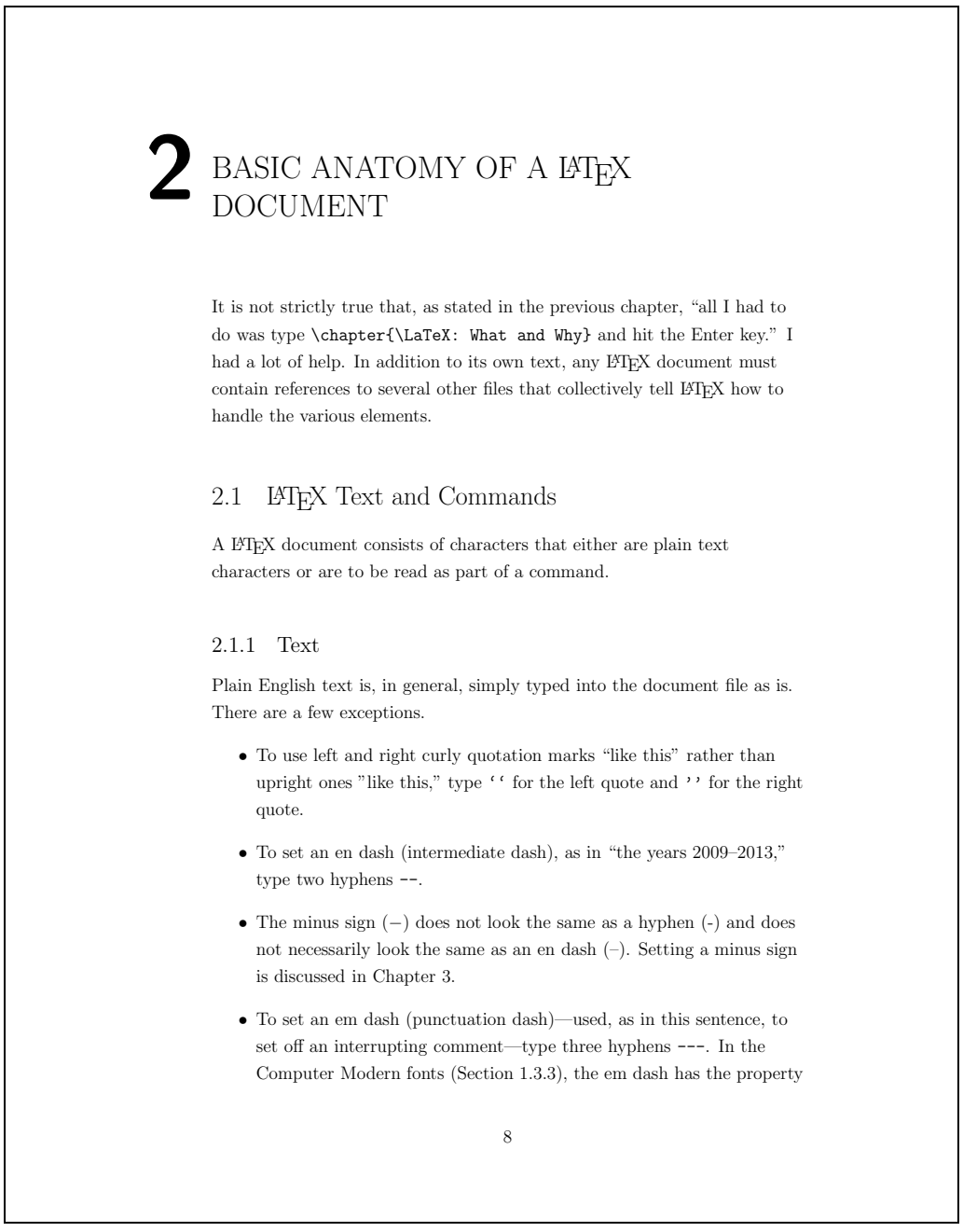
Plain English text is, in general, simply typed into the document file as is. There are a few exceptions.

- To use opening and closing curly quotation marks "like this" rather than two closing ones "like this" (a common error among LaTeX novices!), type ` `` ` for the left quote and `''` for the right quote.
- To set an en dash (intermediate dash), as in "the years 2009–2013," type two hyphens `--`.
- The minus sign (−) does not look the same as a hyphen (-) and does not necessarily look the same as an en dash (–). Setting a minus sign is discussed in Chapter ??.

9

Figure 2.2: Chapter opener with `12pt`, `fullpage`, and `offcenter`.

.



2 BASIC ANATOMY OF A LaTeX DOCUMENT

It is not strictly true that, as stated in the previous chapter, "all I had to do was type `\chapter{\LaTeX: What and Why}` and hit the Enter key." I had a lot of help. In addition to its own text, any LaTeX document must contain references to several other files that collectively tell LaTeX how to handle the various elements.

2.1 LaTeX Text and Commands

A LaTeX document consists of characters that either are plain text characters or are to be read as part of a command.

2.1.1 Text

Plain English text is, in general, simply typed into the document file as is. There are a few exceptions.

- To use left and right curly quotation marks "like this" rather than upright ones "like this," type `` for the left quote and '' for the right quote.
- To set an en dash (intermediate dash), as in "the years 2009–2013," type two hyphens `--`.
- The minus sign (−) does not look the same as a hyphen (-) and does not necessarily look the same as an en dash (–). Setting a minus sign is discussed in Chapter 3.
- To set an em dash (punctuation dash)—used, as in this sentence, to set off an interrupting comment—type three hyphens `---`. In the Computer Modern fonts (Section 1.3.3), the em dash has the property

8

Figure 2.3: Chapter opener with `12pt`, `fancy`, `offcenter`, and ragged right margin produced by removing the `noragright` option.

2.4.5 Controlling margin alignment

When `uiucecethesis09` was written, the Graduate College required margins to be ragged right rather than justified (lined up neatly, as here), apparently to avoid loosely spaced lines and hyphenation problems encountered by non-TEX users. However, the input file ecethesis.tex overrides this default with the option `noragright` so that text is justified as TEX was designed to do. The Graduate College no longer requires ragged right.

2.4.6 Controlling the depth of Table of Contents entries

The option `tocnosub` is new with the `uiucecethesis09` class to prevent subsections and subsubsections from generating lines in the Table of Contents, as discussed in Section 2.8.

2.4.7 Controlling chapter title capitalization

The `mixcasechap` option allows chapter titles to be set in Mixed Case instead of ALL CAPS, as described in Section 2.10.2. Also note that you *must* use the `mixcasechap` option if you are also using the Hyperref package for hyperlinks. Hyperref is incompatible with the all caps default of the `uiucecethesis09` class.

2.4.8 Generating the ProQuest abstract

The `proquest` option is used to produce the ProQuest abstract required with a doctoral dissertation, as described in Section 2.7.

2.5 The Document Body

After the preamble comes the body of the document. All of the text and formatting instructions are located between a command `\begin{document}` and another commmand `\end{document}`.

2.5.1 Document structure elements

LaTeX automatically provides support for dividing a document into chapters, sections within chapters, subsections within sections, and subsubsections within subsections. It should not come as a surprise that these divisions begin with the `\chapter`, `\section`, `\subsection`, and `subsubsection` commands, respectively. As was shown in Section 1.2 for the `\section` command, these commands take the title of the structure element as argument and automatically format the chapter opener (section header, etc.) as prescribed by the document class file; assign a unique sequential number to the chapter (section, etc.), appended to the numbers of all the higher-level divisions in which it is placed; and add a line to the table of contents.

A common mistake is to use the `\chapter` command in the `ecethesis.tex` starter file, rather than (or in addition to) in the separate chapter file where the command belongs, with the unsatisfactory result that the chapter title will appear on a separate page before the chapter text.

2.5.2 Cross-references

One of the most powerful features of LaTeX is its support for automatically cross-referencing numbered elements—chapters and other structural divisions, equations, figures, and tables—using the `\label` and `\ref` commands.

The `\label` command, placed after a `\chapter`, `\section`, or other command starting a structural division,

```
\subsection{Cross-references}\label{subsec-xref}
```

associates the name given as argument of the `\label` command with that division. Then, wherever a `\ref` command occurs that has the same name as its argument, e.g.,

```
This is Section \ref{subsec-xref}.
```

the complete chapter, section, or other division number is printed, as follows: This is Section 2.5.2.

The great benefit of this feature is that if another section (subsection, etc.) is inserted above the one being referred to, the use of a `\ref` to a `\label` instead of writing the number directly in the file ensures that the section (subsection, etc.) number is always correct in the printed file.

Figures (Chapter 4), tables (Chapter 5), and numbered equations (Chapter 3), which are all environments, can also have `\label` commands inside them. For example, Equation (1.1) contains the label `\label{eq:maxdivE}`. Anywhere you want to refer to that equation, you put a `\ref` command whose argument is the same as the argument of the `\label` command: `Equation (\ref{eq:maxdivE})` produces "Equation (1.1)". (Prefixes such as `subsec-` or `eq:` are not mandatory for labels but are good practice to help keep in mind the document's logical structure and the type of element being referenced.)

There is one wrinkle to this feature. To translate the labels into numbers, LaTeX needs to build a list of labels as it is run, but at the same time it also has to fill in the numbers, which it can only do using a *previous* label list. So if you add, delete, or move numbered elements and thereby change the numbering, you will need to run LaTeX *twice* to get the numbers to come out correctly.

2.5.3 Frontmatter and endmatter

In addition to the elements discussed in Section 2.5.1, LaTeX provides support for automatically producing frontmatter and endmatter elements such as title pages, tables of contents, and lists of tables and figures. The exact range of frontmatter and endmatter elements depends in part on the document class.

Title pages require the document title, the author's name, and perhaps other information, depending on the document class. These pieces of information are input using commands such as `\title`, `\author`, and so on. After all these commands, the title page is actually inserted by the `\maketitle` command. The specific pieces of information required for an ECE thesis or dissertation title page are described in Section 2.6.

LaTeX builds the table of contents by recording the `\chapter`, `\section`, `\subsection`, and `\subsubsection` commands as it encounters them. There are also commands to force input into the table of contents: `\addcontentsline` to create an entry equivalent to a chapter, section, etc., and `\addtocontents` to add miscellaneous material. The table of contents itself is actually inserted into the document with the command `\tableofcontents`. The format of the table of contents, and which entries it actually includes, are determined by

the document class file. If the table of contents changes (e.g., by adding, deleting, or moving chapters or sections), LaTeX has to be run twice to put a correct table of contents in the output, just as with cross-references.

Frontmatter elements such as prefaces and forewords (not generally found in theses and dissertations) usually occur *before* the table of contents and are not listed in it. They do not have chapter numbers, but otherwise they are generally formatted like chapters. To start such an element, use the `\chapter*` command:

```
\chapter*{Preface}
```

The asterisk means that LaTeX will not generate a chapter number, will not set a chapter number in the heading, and will not create a table of contents entry for the preface.

LaTeX builds a list of numbered tables and a list of numbered figures by recording the occurrences of the `table` and `figure` environments. To place these lists in the document, the `\listoftables` and `\listoffigures` commands may be used. With the `uiucecethesis09` class, these commands automatically insert entries in the table of contents for the lists (though they do not do so in LaTeX generally).

```
\listoffigures
\listoftables
```

Even though LaTeX automatically maintains the lists of figures and tables, it is not mandatory to include them in the document. If you do not want a list of tables or a list of figures, just omit the `\listoftables` command, the `\listoffigures` command, or both.

Similar commands are used to insert reference lists or bibliographies (see Chapter 6) and indexes.

2.6 Title Page and Copyright Page

As described in Section 2.5.3, the title page is inserted by the `\maketitle` command after a series of commands that specify the information that goes on the title page. In the `uiucecethesis09` document class, the `\maketitle` command also produces the copyright page. The title page needs the following pieces of information from you:

- The title of the thesis, supplied as the argument to the `\title` command. The title is automatically set in all caps, whether or not it is typed in all caps. To break lines as desired, use `\\`.

- Your name exactly as it appears in the official records. The name must be supplied as the argument to the `\author` command in mixed case (e.g., "Firstname M. Lastname"). It will be automatically set in all caps on the title page and left in mixed case on the copyright page.

- The degree sought. This is specified by using exactly one of the following four commands: `\msthesis`, `\phdthesis`, `\othermasters`, or `\otherdoctorate`. The latter two require the full name of the degree and the abbreviation of the degree as arguments (for example, `\otherdoctorate{Doctor of Medicine}{M.D.}`). This command automatically determines whether the word "THESIS" or the word "DISSERTATION" appears on the title page as well as what is to be done with the adviser and committee names. The title page for a Ph.D. dissertation (`\phdthesis`) was shown in Figure 2.1.

- The year the degree is to be awarded, supplied as an argument to the `\degreeyear` command.

- If you are a doctoral candidate or you have a master's committee, the names of the members of the committee, supplied as the argument to the `\committee` command. Use `\\` to separate the committee members.

- The name of your adviser. This is necessary for all master's candidates except those few who have a committee. Whereas master's candidates must show the adviser's name on the title page, doctoral candidates must show the adviser's name on the ProQuest abstract as stated in Section 2.7. Supply the adviser's name as the argument to the `\adviser` (or `\advisor`) command.

- (An old command, `\schools`, was used to list previous degrees and schools under the old title page requirement, and is commented out.)

You may find it necessary to slightly adjust the file `uiucecethesis09.cls` in order to achieve the required 1.5- and 2-inch indents for the Adviser/Doctoral Committee text block on the title page. Lines 234–235, 238–239, and 241–242

of `uiucecethesis09.cls` contain the horizontal spacing commands (`\hspace`) for these indents, depending on which master's/doctoral options you select:

```
233  \if@doctoral
234    \hspace*{0.1in}Doctoral Committee:\\[12pt] %default is 0.2in
235    \hspace*{0.53in}\begin{tabular}{l}\@committee\end{tabular}%\vfil} %default is 0.6in
236  \else
237    \if@c@mmitt@@
238      \hspace*{0.1in}Master's Committee:\\[12pt] %default is 0.2in
239     \hspace*{0.53in}\begin{tabular}{l}\@committee\end{tabular}%\vfil} %default is 0.6in
240    \else
241    \hspace*{0.1in}Adviser:\\[12pt] %default is 0.2in
242     \hspace*{0.53in}\@advisor%\vfil} %default is 0.6in
```

Displayed/printed results may vary with your selections of font size, master's or doctoral, and PDF, PS, or DVI. Best results for the title page of this document were found to correspond with settings of `0.0in` and `0.4in` in lines 234 and 235, respectively.

2.7 Abstract and ProQuest Abstract

The abstract is contained in the `abstract` environment. Only the text of the abstract needs to be entered. The title "Abstract" is set automatically. Note that the abstract should not cite any references or refer to any numbered element in the thesis.

The supplemental material to be deposited with a doctoral dissertation includes the abstract formatted for publication in *Dissertation Abstracts International* published by ProQuest/UMI. The `proquest` option enables you to produce this abstract automatically. When the dissertation is complete and correct as shown by a copyedit check, set the `proquest` option and run LaTeX on the dissertation to produce the ProQuest abstract, consisting of the ProQuest header over the content in the `abstract` environment. With this option set, no output is produced except the ProQuest abstract. Without the `proquest` option, the whole dissertation, including the abstract formatted as a normal part of the dissertion, is produced.

For the ProQuest abstract, the adviser name is necessary. Set the adviser name using the `\adviser` command described in Section 2.6.

2.8 Table of Contents

The table of contents is automatically generated when LaTeX is run on the thesis, and it is placed in the thesis itself by means of the `\tableofcontents` command, as discussed in Section 2.5.1.

LaTeX ordinarily generates lines in the table of contents for all numbered structural elements down to the subsection level, as shown in Figure 2.4, but the Graduate College requires table of contents entries only at the chapter level. Leaving the lower-level entries out simplifies the thesis check process, so the Graduate College "strongly encourages students to keep the table of contents as simple as possible" [7]. The `tocnosub` option prevents lower-level subsections and subsubsections from being added to the table of contents, as shown in Figure 2.5.

One strict Graduate College requirement that may cause difficulty for LaTeX users is that "Leader dots must be displayed between the final word of each title and the page number" [7]. As shown in Figure 2.6, the title of Chapter 6 of this thesis is just long enough to prevent leader dots, but not long enough to go to a second line where leader dots would then fit. Shortening or lengthening the title is an obvious solution, but one that is unsatisfactory for writers who have thought carefully about their titles. ECE graduate student Mike Daly provided the solution used to achieve the result in the actual Table of Contents of this thesis, the relevant detail of which is reproduced in Figure 2.7. The solution is coded as follows:

```
\chapter[Producing References and \, \, \, \, \, \, \, \, \, Bibliography]
{Producing References and Bibliography}
\label{chap-bib}
```

Note that the `\chapter` function takes two arguments: the first, in square brackets, specifies how you want the title to appear in the table of contents and contains enough thin spaces (`\,`) to force a line break with the resulting leader dots; the second, in curly braces, specifies the title for the text. Make sure the wording is exactly the same for both arguments!

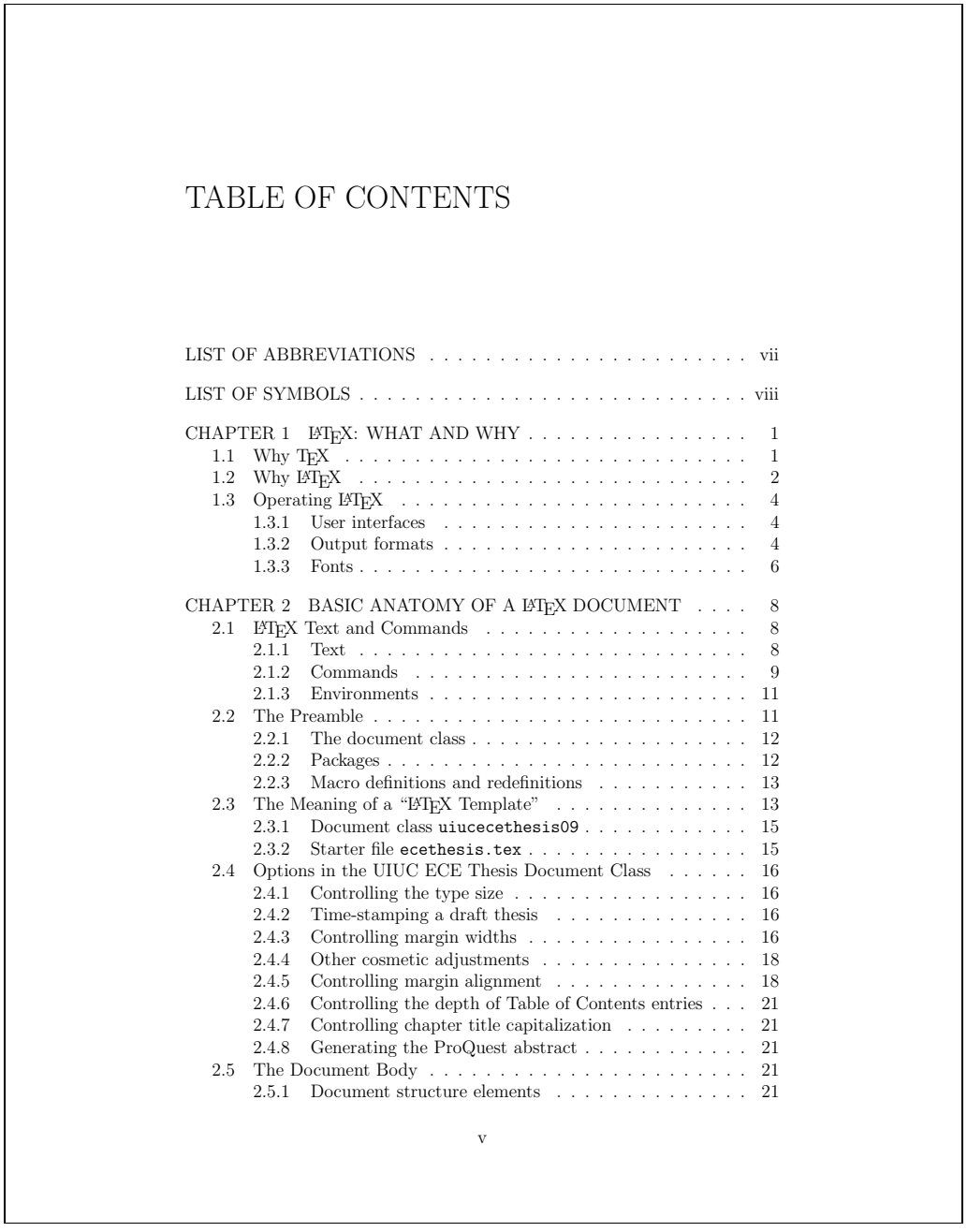


TABLE OF CONTENTS

v

Figure 2.4: Table of contents with subsections listed (not recommended).

TABLE OF CONTENTS

Figure 2.5: Table of contents with subsections suppressed by the `tocnosub` option (recommended).

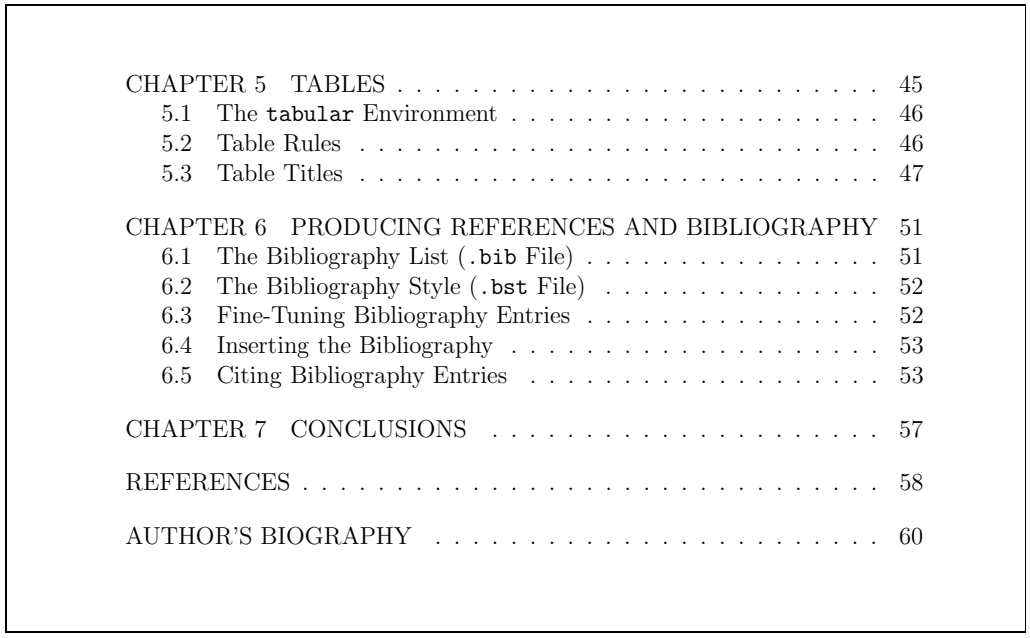


Figure 2.6: Chapter 6 title does not allow for required leader dots.

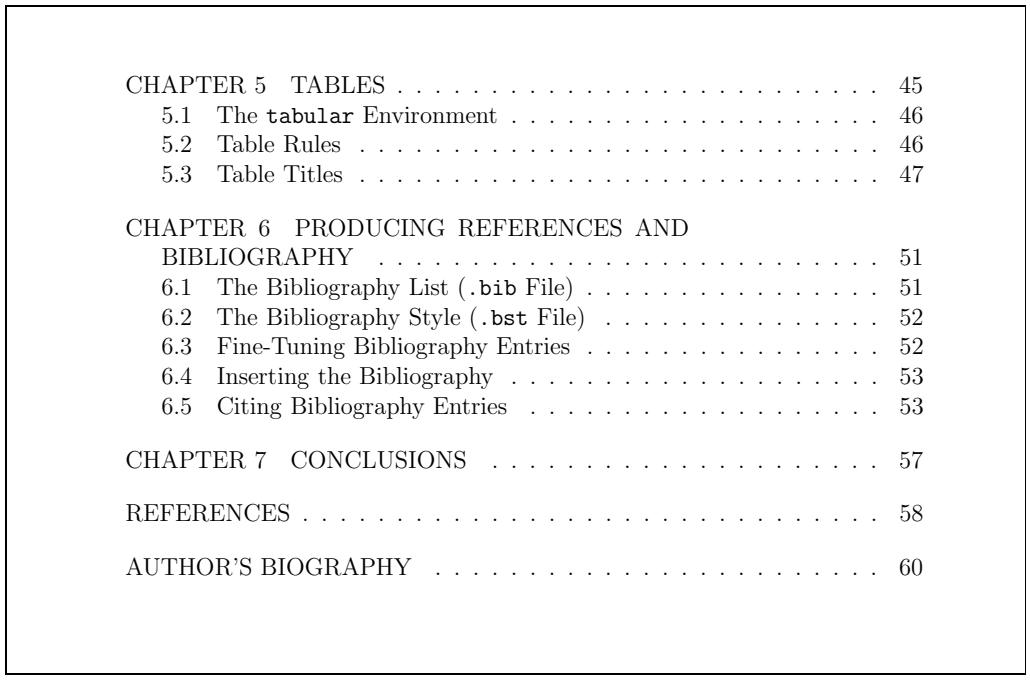


Figure 2.7: After Mike Daly's fix, Chapter 6 title accommodates required leader dots.

2.9 Lists of Tables, Figures, Symbols, and Abbreviations

LaTeX provides support for generating a list of tables and a list of figures automatically. These lists are placed in the thesis by means of the `\listoftables` and `\listoffigures` commands. These commands are included in the starter file. The Graduate College and the ECE Department do not require these lists, and leaving them out simplifies the Publications Office review of the thesis. Therefore, you may wish to delete these commands or comment them out, especially if you are submitting on deadline.

The template provides the `symbollist` environment to support lists of symbols and abbreviations, and the commands for them are included in the starter file, but if you use such lists, you must make the entries by hand. Otherwise, just comment out the `\chapter{List of Abbreviations}` command.

2.10 The Chapters

2.10.1 Writing chapters in separate files

A large project such as a thesis or a book can be difficult to manage. If the entire project is stored in a single file, the slightest change can require a needle-in-haystack search through the whole file. Keeping the project in multiple source files—such as one file for each chapter and one file that provides the preamble, title page, frontmatter, and endmatter—can help keep the complexity under control.

The command `\include` makes this division into multiple source files possible. It inserts the contents of the text file whose file name is given by the argument to the `\include` command and whose extension is `.tex`. The file that contains the preamble and the code for the title page and so forth can therefore contain `\include` commands to bring in the individual chapter files, e.g.,

```
\include{introduction}
\include{theoretical}
```

```
\include{priorwork}
\include{expdesign}
\include{results}
\include{discussion}
\include{conclusion}
```

to include the individual chapter files `introduction.tex`, `theoretical.tex`, and so on.

2.10.2 Chapter title capitalization

As described in Section 2.5.1, the chapter title is the argument to the `\chapter` command. The `uiucecethesis09` class now automatically converts the chapter titles to uppercase by default, whether they are entered as ALL CAPS, Title Case (headline style), or Sentence case (sentence style). If you prefer to use Title Case in your chapter titles, use the `mixcasechap` option in your `\documentclass` command. If you use this option, the chapter titles will appear exactly as you enter them. You are then responsible for correctly capitalizing the first letter of "all important words" [8]. According to one well-respected source [9], important words are the first and last words of the sentence and all other words except "a," "an," "the"; "and," "or," "but," "for," "nor"; and prepositions. Title Case is essential if any of your chapter titles contains a symbol that requires lowercase letters, such as "GaAs."

2.11 A Few Special Commands

2.11.1 Blank lines

No matter how many consecutive line spaces you put in your source file between paragraphs, LaTeX will produce only one paragraph break to stand for all of them. To add a blank line in your output, the `\extraline` command is provided with the `uiucecethesis09` document class (see Sections 2.2.1 and 2.3.1). Place this command at the end of the paragraph before the desired location of the blank line.

2.11.2 Verbatim (code) material

A special command `\verb` is used to produce the command examples verbatim in typewriter-like font in this thesis. The material to be quoted verbatim may contain backslashes and curly braces itself, so these characters cannot be used to mark the beginning and end of the argument. Instead, the first character after `\verb` must be a character not found in the verbatim material, and then the same character marks the end of the verbatim material: ``\verb|\TeX|,'' produces "\TeX," which in turn produces "TEX," as you know already.

For displays of verbatim material, such as computer programs,

```
for (i=0; i<500; i++) {
  printf("I will not automate my punishment assignments.\n")
}
```

LATEX provides the `verbatim` environment:

```
\begin{verbatim}
for (i=0; i<500; i++) {
  printf("I will not automate my punishment assignments.\n")
}
\end{verbatim}
```

The `\verbatim` environment ends with the first `\end{verbatim}` command encountered. To produce the preceding display it was necessary to embed each line from `\begin{verbatim}` through `\end{verbatim}` in a structure `\noindent\verb|...|\\`.

2.11.3 Suppressing part of the text

You may wish to prevent part of your thesis from printing during the course of editing it. For example, you may want to work in one chapter at a time and leave the rest of them out so that you will not have to page through the other chapters in the output to get to the chapter in which you are interested. One way of preventing printing is to comment out the unwanted lines with the `%` character:

```
%\include{introduction}
\include{theoretical}
%\include{priorwork}
%\include{expdesign}
%\include{results}
%\include{discussion}
%\include{conclusion}
```

The `uiucecethesis09` class, however, also provides a command `\ignore`, which you can use instead:

```
\ignore{
\include{introduction}
}
\include{theoretical}
\ignore{
\include{priorwork}
\include{expdesign}
\include{results}
\include{discussion}
\include{conclusion}
}
```

You can also use the `\ignore` command in-line:

```
he is \ignore{not }the sharpest knife in the drawer
```

produces "he is the sharpest knife in the drawer." Note that one of the surrounding spaces must be included in the `\ignore` argument, because otherwise an extra space is produced:

```
he is \ignore{not} the sharpest knife in the drawer
```

produces "he is the sharpest knife in the drawer."

CHAPTER 3

MATH

As mentioned at the beginning of this document, TEX was optimized for math from its beginning. A special *math mode* enables many mathematical symbols to be set and correctly formatted as in the best mathematical textbooks.

3.1 Math within a Paragraph

For math within a paragraph, a dollar sign `$` is placed at the beginning and at the end of what is to be math: `$\pi r^2$` sets πr^2. (To set a dollar sign, in math mode or out, precede it with a backslash: `\$` produces \$, `$\$2000.00x$` produces $\$2000.00x$.) The environment `math` does the same: `\begin{math}` $\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$ `\end{math}` was produced with `\begin{math}` and `\end{math}` commands, right inside the paragraph itself.

Some common notations that may not appear to be particularly mathematical nevertheless require LATEX's math features to produce.

3.1.1 Special characters

As described in Section 2.1.2, the backslash character in ordinary font requires math mode: `$\backslash$` produces \. (As described in Section 2.11.2, you can get a backslash in typewriter font with the `\verb` command: `\verb|\|` produces \.)

3.1.2 Minus and plus signs on numbers

Although plain numbers do not ordinarily require math mode (although math mode is recommended for numbers for consistency's sake if you are using a

font other than Computer Modern for basic text), negative numbers should be set in math mode to make the minus sign come out correctly: `$-3$` to get -3 rather than -3. If you have both numbers with positive signs and those with negative signs, both should be written using math: `$+3$` produces $+3$.

3.1.3 Subscripts and superscripts

All superscripts and subscripts (other than note citations) should be produced using math mode. Put a caret `^` before a symbol to be represented as a superscript and an underscore `_` before a symbol to be represented as a subscript: `$mc^2$` produces mc^2, `$e_y$` produces e_y.

Multiple-character superscripts and subscripts need to be enclosed in curly braces: `$e^{-j\omega t}$` produces $e^{-j\omega t}$. If the curly braces are not used, only the first character will be super- or subscripted: `$e^-j\omega t$` produces $e^-j\omega t$.

It is not necessary for the symbol next to the subscript or superscript to be in math mode for the superscripting or subscripting to work. For example, `TM$_{11}$` produces TM_{11}. The superscript or subscript does not even have to follow a symbol; for example, `$^{14}$C` produces ${}^{14}\mathrm{C}$.

If superscripts and subscripts occur together, LaTeX will stack superscripts over subscripts, as in SO_4^{2-} (produced by `SO$_4^{2-}$`) or ${}_2^4\mathrm{He}$ (produced by `$^4_2$He`). If it is necessary to offset the subscript and superscript, enclose the first element in curly braces: ${g_a}^b$ (produced by `${g_a}^b$`), ${g^a}_b$ (produced by `${g^a}_b$`).

3.1.4 A word about multiletter symbols

As shown by the examples at the beginning of this chapter, putting a letter in math mode italicizes it, while putting a number in math mode leaves it upright: `$a^2+b^2=c^2$` produces the ordinary algebraic equation $a^2 + b^2 = c^2$.

Multiple consecutive letters are set as if they represented single-letter variables that are multiplied together to make a term: `$ax^2+bx+c=0$` produces $ax^2 + bx + c = 0$. Spaces between such letters are ignored: the command `$a x^2 + b x + c = 0$` produces $ax^2 + bx + c = 0$.

Some writers prefer to write equations that contain whole words as variable names for clarity rather than using single letters. If such variable names are simply entered in math mode, the spacing between the letters is all wrong for a word, and any spaces in the variable name are lost: `$y=x+Difference between the two terms$` produces the hideous expression

$$y = x + Differencebetweenthetwoterms$$

To avoid the problem, put the multicharacter variable name into the argument of a `\mathit` command if you want it to appear in italic. You will also need to force spacing (backslash-space, `\ `): `$y=x+\mathit{Difference\ between\ the\ two\ terms}$` produces

$$y = x + \mathit{Difference\ between\ the\ two\ terms}$$

Often in textbooks, non-italic letters are used for a multiletter variable name for clarity's sake. In that case, use `\mathrm` (math roman, i.e., not italic) rather than `\mathit`. For example, the command `$y=x+\mathrm{Difference\ between\ the\ two\ terms}$` produces

$$y = x + \mathrm{Difference\ between\ the\ two\ terms}$$

The latter remark also holds for explanatory subscripts. In textbooks, when a multiletter subscript means a word, it is often set roman, whereas if a subscript consists of a combination of individual letters that each have their own meaning, they are set italic: `$V_{GS}$` produces V_{GS}, meaning the voltage (V) between the gate (G) and the source (S), but `$t_\mathrm{off}$` produces t_{off}, representing the time (t) during which the device is off.

3.2 Display Math

The `displaymath` environment produces math formatted to be displayed in its own vertical space:

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

was produced by the same math coding as in the first paragraph of Section 3.1, but with the `displaymath` environment invoked instead of `math`. Notice that when the equation is inline, the fraction is set small enough to fit on the line, whereas in display math the fraction is set full size. (To get the fraction, the command `\frac` is used, followed by the numerator and denominator expressions, each in a pair of braces.)

3.3 Numbered Equations

Numbered equations such as Equations (1.1)–(1.4) are produced by using the `equation` environment for a single equation, or the `eqnarray` environment for multiple equations with their equal signs lined up, rather than the `\displaymath` environment. To get the equations to line up on their equal signs, the special alignment character `&` is placed on either side of the equal sign. These environments put an equation number at the right of every line. To get a similar alignment of equations but without equation numbers, the `eqnarray*` environment is used.

```
\begin{eqnarray*}
\textbf{D}&=&\varepsilon\textbf{E}\\
\textbf{B}&=&\mu\textbf{H}
\end{eqnarray*}
```

produces

$$\begin{aligned} \mathbf{D} &= \varepsilon\mathbf{E} \\ \mathbf{B} &= \mu\mathbf{H} \end{aligned}$$

3.4 Arrayed Elements Inside Equations

The `eqnarray` environment should not be confused with the `array` environment, which produces vertical alignments *inside* an equation, such as for use in representing vectors, matrices, and determinants.

```
\begin{displaymath}
\nabla\times\textbf{E}=\left|
\begin{array}{ccc}
\textbf{e}_x&\textbf{e}_y&\textbf{e}_z\\
\frac{\partial}{\partial x}&
\frac{\partial}{\partial y}&
\frac{\partial}{\partial z}\\
E_x&E_y&E_z\\
\end{array}
\right|
\end{displaymath}
```

produces

$$\nabla \times \mathbf{E} = \left| \begin{array}{ccc} \mathbf{e}_x & \mathbf{e}_y & \mathbf{e}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ E_x & E_y & E_z \end{array} \right|$$

The `array` environment has to be invoked inside math mode, whereas the `math`, `displaymath`, `equation`, `eqnarray`, and `eqnarray*` environments have to be invoked outside of math mode.

Inspection of the LaTeX code for the preceding equation shows a few more features of interest. The second argument to the `\begin{array}` command indicates the left (`l`), centered (`c`), or right (`r`) alignment of the array columns. The `\left` and `\right` commands make the chosen delimiters (parentheses, brackets, braces, or vertical lines) tall enough to enclose whatever is between them. For many more of LaTeX's math features, see Chapter 3 of [5]. The TeXnicCenter IDE has toolbar buttons to insert codes for math symbols with a mouse click.

3.5 Limits, Integrals, Summations

Integrals, limits, summations, and the like have expressions above and below them. These are handled like subscripts and superscripts. Thus,

```
\int_a^b f(x)dx =
\lim_{\Delta x\rightarrow 0}
\sum_{i=0}^{(b-a)/\Delta x} f(a+i\Delta x)\Delta x
```

produces

$$\int_a^b f(x)dx = \lim_{\Delta x \to 0} \sum_{i=0}^{(b-a)/\Delta x} f(a + i\Delta x)\Delta x$$

in display mode and $\int_a^b f(x)dx = \lim_{\Delta x \to 0} \sum_{i=0}^{(b-a)/\Delta x} f(a + i\Delta x)\Delta x$ inline.

3.6 Tables of Symbols

Tables of all the math symbols available in LaTeXare available through CTAN [3] on the path `tex-archive/info/symbols/math/`.

CHAPTER 4

FIGURES

As can be inferred from Figure 1.1, LaTeX supports placement of figures in documents, automatically giving them numbers that can be cross-referenced and placing them at the top or bottom of the page as requested by the user. The `figure` environment is used to produce a figure. For example, Figure 4.1 shows a simple graphic containing text and was included by means of the following code:

```
\begin{figure}[tbp]
  \includegraphics{tocreate.eps}
  \caption{A simple graphic.}
  \label{fig:cclogo}
\end{figure}
```

4.1 Elements of the `figure` Environment

Like sections, subsections, and equations, figures are numbered elements (unnumbered figures can be produced with the `figure*` environment). As with those elements, LaTeX assigns the figure numbers as it runs, but it puts the numbers in the file based on the previous set of assignments, so after a figure is added, deleted, or moved, LaTeX must be run twice.

Figure 4.1: A simple graphic.

4.1.1 Including a graphics object

In the `figure` environment, the `\includegraphics` command imports the graphics file itself. In this case, the file is an Encapsulated PostScript (EPS) file, `tocreate.eps`. This file type is one of the easiest to incorporate into a LaTeX document. Other types of graphics files, such as those representing photographs, screen shots, or scanned images, can also be incorporated. For example, the file used for Figure 1.1 is a JPEG (Joint Photographic Experts Group) compressed raster image.

4.1.2 Setting the caption

The `\caption` command places the caption above or below the figure, depending on whether it precedes or follows the `\includegraphics` command, and causes LaTeX to generate a figure number and print the figure number and the text of the argument, just as the `\section` command generates a new section number and prints it out along with the section title passed as the argument.

4.1.3 Setting the label

The `\label` command tells LaTeX to equate the label with the figure number generated with the `\caption` command so that a `\ref` command will print the figure number. If the `\caption` command is omitted, a `\ref` to the label will produce the most recent section or subsection (or subsubsection) number instead of a figure number.

4.2 The Bounding Box

The following code was used for setting Figure 1.1:

```
\begin{figure}[tb]
\resizebox{\textwidth}{!}{
  \includegraphics[0pt,0pt][966pt,773pt]{texniccenter.jpg}
}
\caption{The {\TeX}nicCenter user interface.}
```

```
\label{fig:texniccenter}
\end{figure}
```

The `\includegraphics` command for this figure is much more complicated than for Figure 4.1. The reason is that LaTeX needs to know the location of the boundaries of the image, known as its *bounding box*. An EPS file such as that used for Figure 4.1 already contains the bounding box information in a form that LaTeX can recognize. A JPEG file such as that used for Figure 1.1, however, does not, so the user has to state the horizontal and vertical coordinates of the bottom left and top right corners of the bounding box as optional arguments to the `\includegraphics` command.

There are a couple of wrinkles to stating the bounding box. One is that for a JPEG file, LaTeX interprets the numbers of *points* (`pt`, nominally $\frac{1}{72}$ inch) in the bounding box dimensions as numbers of *pixels*, no matter at what resolution the image was recorded. Adobe Photoshop reports a resolution of 96 pixels per inch for the image used for Figure 1.1, but LaTeX knew nothing of it.

Another wrinkle is that to include the entire image, only the top right corner of the bounding box actually needs to be stated. For example,

```
\includegraphics[966pt,773pt]{texniccenter.jpg}
```

would have worked equally well for Figure 1.1.

4.3 Controlling the Figure Size

At the resolution at which the screen shot in Figure 1.1 was made, the image would have been over 10 inches wide, too large to fit on the page. To shrink the image to fit between the margins allowed by the Graduate College, the whole `\includegraphics` command is placed inside a `\resizebox` command as its third argument, as seen in Section 4.2. The first argument to that command is the width to which the figure is to be sized; the value `\textwidth` shrinks the figure automatically to exactly the width between the margins. The second argument to `\resizebox` is the height to which the figure is to be resized; using `!` here preserves the original proportions of the figure without the user having to do calculations.

4.4 Cropping a Figure

By stating nonzero values for the bottom left corner and placing an asterisk after `\includegraphics`, a figure can be cropped. For example,

```
\begin{figure}[bp]
  \begin{center}
    \includegraphics*[98.5pt,4pt][160pt,37pt]{tocreate.eps}
  \end{center}
  \caption{Cropping the graphic in Figure \ref{fig:cclogo}.}
  \label{fig:cropcclogo}
\end{figure}
```

produces Figure 4.2. Note the `center` environment, without which the cropped figure would be set flush left while the caption is centered—a gaffe that was committed in Figure 4.1, but that would be much more noticeable here because the image is narrower. Note also that the cropping values need not be integers.

4.5 Borders on Figures

You can set borders around figures, as was done for the page images in Chapter 2, by using the `\framebox` command around the `\includegraphics`. The following code produces Figure 4.3.

```
\begin{figure}[bp]
  \begin{center}
    \framebox{
      \includegraphics*[98.5pt,4pt][160pt,37pt]{tocreate.eps}
    }
  \end{center}
  \caption{Framing the graphic in Figure \ref{fig:cropcclogo}.}
```

Figure 4.2: Cropping the graphic in Figure 4.1.

```
  \label{fig:framecclogo}
\end{figure}
```

Notice that the frame is some distance outside the cropped graphic itself. Therefore, if you use the `\resizebox` command to constrain the figure within the text margins, you must place the `\framebox` inside the third argument to `\resizebox`, not the other way around. Otherwise, you will get "Overfull `\hbox`" messages when you run TEX, and you will also exceed the Graduate College margins.

4.6 Controlling Figure Position

The optional argument to the `\begin{figure}` command consists of a selection among `t`, `b`, `p`, and `here`. These options specify the possibilities for placement of the figure, which are, respectively: at the top of the current page, at the bottom of the current page, on a page of its own, or as close as possible to where the `figure` environment appears in the source file, no matter where on the page that may be. If all three of the letter choices (`t`, `b`, `p`) are entered, LATEX will place the figure at what it determines to be the best location, which may not be where the user wants it. By changing the selection of these options or by movement of the `figure` environment itself, the user can control the placement of the figure.

For example, for Figures 4.2 and 4.3, the code for the `figure` environments was placed in the file immediately after the figures were first mentioned in text. For these figures the `t` was left out, because if it were left in, LATEX would have put the figures on the top of their pages, before they were first mentioned, which is not what the user wanted. (Another way to achieve the same result would have been to move the `figure` environments farther down in the source file.)

On the other hand, Figure 1.1 (discussed in Section 4.2) was tall enough

Figure 4.3: Framing the graphic in Figure 4.2.

that LaTeX could not set the figure at the top of the page on which its code occurred—if it did, the code would be forced onto the next page. Therefore, the `pb` options caused LaTeX to put that figure on its own page on the next page instead.

The Graduate College encourages placement of all figures at the end of their chapters. To do so, move the `figure` environments. Make sure that the figures are mentioned in consecutive numerical order in the text, and that they appear in that order at the end of the chapter. If you prefer to intersperse your figures in the chapters using the four placement commands and portable `figure` environment, ECE Publications will check that each figure appears on same page where it is first mentioned in text *or* the first possible page after that (which may be much later if a group of figures is referenced in a short passage). Exceptions to this rule may be made, depending on the context.

4.7 Packages for Figures

The `graphicx` package was used to support the figures shown in this document. Other packages that can be used with graphics include

- `epsfig`, which is well suited for manipulating EPS graphics and is described in Chapter 11 of [6]
- `subfigure`, which enables figures with multiple lettered parts that can be manipulated individually

LaTeX itself has some line graphics drawing capabilities, which are described in Chapter 10 of [6].

CHAPTER 5

TABLES

Like figures, tables have a special environment, `table`, which takes care of the table placement at the top of the current page, the bottom of the current page, or a new page; sets up a table number (if a `\caption` command is used); and enables cross-referencing the table (if a `\label` command is used). For example, Table 5.1 is produced by the code

```
\begin{table}[bp]
  \centering
  \caption{Electricity and Magnetism Compared}
    \begin{tabular}{lcc}
&\textbf{Electricity}&\textbf{Magnetism}\\
Field intensity&$\textbf{E}$&$\textbf{H}$\\
Flux density&$\textbf{D}$&$\textbf{B}$\\
Charge density&$\rho$&$\rho_m$?\\
Current density&$\textbf{J}$&$\textbf{J}_m$?
    \end{tabular}
  \label{tab:elmag}
\end{table}
```

Like sections, subsections, and equations, tables are numbered elements (unnumbered tables can be produced with the `table*` environment). As with those elements, LaTeX assigns the numbers as it runs, but it puts the

Table 5.1: Electricity and Magnetism Compared

	Electricity	**Magnetism**
Field intensity	**E**	**H**
Flux density	**D**	**B**
Charge density	ρ	ρ_m?
Current density	**J**	$\mathbf{J}_m$?

numbers in the file based on the previous set of assignments, so after a table is added, deleted, or moved, LaTeX must be run twice.

As with figures, you may wish to group all the tables at the end of their chapters.

5.1 The `tabular` Environment

The heart of the table is the `tabular` environment. The column alignments are specified by the second argument to `\begin`, the columns are separated by `&`, and the rows are separated by `\\`, as in the `array` and `eqnarray` commands discussed in Chapter 3.

5.2 Table Rules

Table 5.1 is not very easy to read as it stands. It would be much more readable if there were lines, called *rules* in publishing, to separate the table from the surrounding material and to separate the column headers from the table body, as in Table 5.2.

The horizontal lines in Table 5.2 were produced by the `\hline` command:

```
\begin{table}[bp]
  \centering
  \caption{Electricity and Magnetism Compared}
    \begin{tabular}{lcc}\hline\hline
&\textbf{Electricity}&\textbf{Magnetism}\\ \hline
Field intensity&$\textbf{E}$&$\textbf{H}$\\
Flux density&$\textbf{D}$&$\textbf{B}$\\
Charge density&$\rho$&$\rho_m$?\\
```

Table 5.2: Electricity and Magnetism Compared

	Electricity	**Magnetism**
Field intensity	$\mathbf{E}$	$\mathbf{H}$
Flux density	$\mathbf{D}$	$\mathbf{B}$
Charge density	ρ	ρ_m?
Current density	$\mathbf{J}$	$\mathbf{J}_m$?

```
Current density&$\textbf{J}$&$\textbf{J}_m$?\\ \hline
    \end{tabular}
  \label{tab:elmagrule}
\end{table}
```

To get vertical lines between table columns, you put vertical bars `|` between the column alignment letters in the argument to the `\begin{tabular}` command as in Table 5.3, produced by

```
\begin{table}[bp]
  \centering
  \caption{Electricity and Magnetism Compared}
    \begin{tabular}{l|c|c}\hline\hline
&\textbf{Electricity}&\textbf{Magnetism}\\ \hline
Field intensity&$\textbf{E}$&$\textbf{H}$\\
Flux density&$\textbf{D}$&$\textbf{B}$\\
Charge density&$\rho$&$\rho_m$?\\
Current density&$\textbf{J}$&$\textbf{J}_m$?\\ \hline
    \end{tabular}
  \label{tab:elmagvrule}
\end{table}
```

5.3 Table Titles

A table title should be placed above the table body, whereas a figure caption should be placed below the figure. Therefore, place the `\caption` command above the `tabular` environment. This does not happen automatically. If you are using TEXnicCenter, for example, and you use Ctrl+Alt+T to insert tabular material, a `table` environment is inserted in which the `\caption` command is below the space allowed for the `tabular` environment,

Table 5.3: Electricity and Magnetism Compared

	Electricity	**Magnetism**
Field intensity	$\mathbf{E}$	$\mathbf{H}$
Flux density	$\mathbf{D}$	$\mathbf{B}$
Charge density	ρ	ρ_m?
Current density	$\mathbf{J}$	$\mathbf{J}_m$?

Table 5.4: Electricity and magnetism compared. The flux density and field intensity are related to each other by the constitutive relations; the flux density is equal to the field intensity times the constitutive factor. The term ε is known as electric permittivity, and μ is known as magnetic permeability. The quantities marked with question marks in the Magnetism column refer to magnetic monopoles, which may or may not exist.

	Electricity	**Magnetism**
Field intensity	$\mathbf{E}$	$\mathbf{H}$
Flux density	$\mathbf{D}$	$\mathbf{B}$
Constitutive factor	ε	μ
Charge density	ρ	ρ_m?
Current density	$\mathbf{J}$	$\mathbf{J}_m$?

```
\begin{table}

\caption{Title of the table}
\label{tab:TitleOfTheTable}
\end{table}
```

and you must move it yourself.

Because figure captions and table titles are both produced by the `\caption` command, one might be inclined to write table titles the same way as one writes figure captions, including as much additional information as the reader would need to understand the symbols and notations in the table body, as in Table 5.4. Because the table title is above the table body, however, this kind of caption is unwieldy to read. If you look at tables in a textbook, you will usually find that the titles are quite short, just a single phrase, and explanatory material is put into notes at the bottom of the table. You can do this by putting the footnote inside the `table` environment below the `tabular` material, as in Table 5.5. In that table, to improve readability even more, the explanatory material has been broken into two notes, and superscript note letters have been used to tie the notes to the table entries to which they apply. Also note that, because it consists of many sentences, the title of Table 5.4 must be set in "sentence case" with capitalization only at the beginning of a sentence and a period at the end. But my preferred style elsewhere in this thesis is the classic "headline style" for table titles, with capitalization for each important word and no punctuation. The solution in Table 5.5 takes care of this problem, too.

Table 5.5: Electricity and Magnetism Compared

	Electricity	**Magnetism**
Field intensity[a]	**E**	**H**
Flux density[a]	**D**	**B**
Constitutive factor[a]	ε[b]	μ[c]
Charge density	ρ	ρ_m?[d]
Current density	**J**	$\mathbf{J}_m$?[d]

[a]The flux density and field intensity are related to each other by the constitutive relations; the flux density is equal to the field intensity times the constitutive factor.
[b]ε is known as electric permittivity.
[c]μ is known as magnetic permeability.
[d]The quantities marked with question marks refer to magnetic monopoles, which may or may not exist.

The following code was used to produce Table 5.5. Note that instead of the `\centering` command used in the code for Table 5.1, the table body is placed in a `center` environment, and the footnotes are placed after the end of that environment with a special instruction to reduce their font size. The `\label` command must be inside the `\center` environment along with the `tabular` material; otherwise LaTeX will produce a section reference rather than a table reference. There must also be line breaks between the footnotes. Finally, note the small extra space (`\vspace{0.05in}`) added after the caption, as is done for every table in this chapter, for visual effect.

```
\begin{table}[tb]
  \begin{center}
  \caption{Electricity and Magnetism Compared}
   \vspace{0.05in}
    \begin{tabular}{lcc}\hline\hline
&\textbf{Electricity}&\textbf{Magnetism}\\ \hline
Field intensity$^a$&$\textbf{E}$&$\textbf{H}$\\
Flux density$^a$&$\textbf{D}$&$\textbf{B}$\\
Constitutive factor$^a$&$\varepsilon$$^b$&$\mu$$^c$\\
Charge density&$\rho$&$\rho_m$?$^d$\\
Current density&$\textbf{J}$&$\textbf{J}_m$?$^d$\\ \hline
    \end{tabular}
  \label{tab:outertfn}
  \end{center}
```

```
\footnotesize{$^a$The flux density and field intensity are related
to each other by the constitutive relations;
the flux density is equal to the field intensity
times the constitutive factor.

$^b\varepsilon$ is known as electric permittivity.

$^c\mu$ is known as magnetic permeability.

$^d$The quantities marked with question marks refer to
magnetic monopoles, which may or may not exist.}
\end{table}
```

CHAPTER 6

PRODUCING REFERENCES AND BIBLIOGRAPHY

A powerful utility called BibTeX, written by Oren Patashnik, works with LaTeX to produce correctly formatted bibliographies and reference lists and accurate citations in text.

6.1 The Bibliography List (`.bib` File)

The bibliography entries must be listed in a text file with the extension `.bib`. The file name does not have to be the same as the LaTeX document file name (therefore the same bibliography can be used with multiple LaTeX projects). A sample reference list, `thesisrefs.bib`, is supplied as part of the ECE template; you can start with this and add your own entries at the end. (Only those entries cited with `\cite` or `\nocite` will be included in the bibliography itself.)

Here is the entry in the `.bib` file for [6]:

```
@book{LaTeXCompanion,
  author        =
    "Michael Goossens and Frank Mittelbach and Alexander Samarin",
  title         = "The \LaTeX\ Companion",
  publisher     = "Addison-Wesley",
  address       = "Reading, MA",
  year          = "1994"
};
```

The entry begins with `@book`, which specifies what type of bibliography entry this is, and contains a list of pieces of information within a pair of curly braces. The first piece of information is a label to enable the entry to be cited within the main LaTeX document using the `\cite` command with that

label as argument. Thus, `\cite{LaTeXCompanion}` produces "[6]" in the main document. The rest of the information consists of field names, such as `author`, each followed by an equals sign and the information itself, which may include TEX commands, in quotation marks.

6.2 The Bibliography Style (`.bst` File)

Comparing the code for reference [6] in Section 6.1 with that entry in the References, you can see that the software has performed some intelligent operations on the information to make the bibliography entry conform to IEEE style, such as the title in italics followed by a period and extra space and, more impressively, the first names of the authors reduced to initials. This magic is wrought by BIBTEX using a bibliography style file with the extension `.bst`. A LATEX distribution contains an IEEE style file `IEEEtran.bst`, but for UIUC ECE theses a slightly modified form of this file, `IEEE_ECE.bst`, is provided. The style file also directs BIBTEX to translate in-text citations to sequential numbers [1], [2], ..., to give the corresponding numbers to the entries, and to output the entries in the order of these numbers (regardless of the order in which they were entered in the `.bib` file), all in keeping with IEEE style. There are style files to support many different bibliographic conventions, for example, entries sorted and cited by author and year.

6.3 Fine-Tuning Bibliography Entries

Sometimes it is necessary to get around the smart formatting features defined in the `.bst` file. For example, in the entry for [10], the author is an organization, The Unicode Consortium. Processing the field `author = "The Unicode Consortium"` for IEEE style turns this name into "T. U. Consortium." Enclosing the entire organization name in curly braces causes the name to be treated as a unit rather than analyzed into first, middle, and last names:

```
author = "{The Unicode Consortium}",
```

And in the entry for [11], if the article title were not given special treatment, then the unit of measure (GHz), initialism (UV), and proper name (Bragg)

would appear incorrectly as ghz, uv, and bragg. To prevent that result, the whole name is enclosed in curly braces:

```
title = "{150 GHz of thermo-optic tuning in direct
        UV written silica-on-silicon planar Bragg grating}",
```

Alternatively, each individual term requiring an override of BibTeX defaults can be enclosed in curly braces.

6.4 Inserting the Bibliography

The bibliography itself is inserted in the main LaTeX document using the commands `\bibliographystyle`, whose argument is the file name of the `.bst` file, and `\bibliography`, whose argument is the file name of the `.bib` file:

```
\bibliographystyle{IEEEtran} % IEEEtran.bst
\bibliography{references}    % references.bib
```

When LaTeX encounters these two commands, it writes the file names into a file pertaining to the document that it keeps for its own use (where it also keeps labels, numbers, and titles of chapters, sections, tables, figures, and other numbered elements for producing tables of contents, lists of tables and figures, and cross-reference resolutions). The separate program `bibtex` then has to be run on this file to generate the correctly formatted and sorted bibliography and the citations translated into the correct format (an IDE such as TeXnicCenter carries out this step automatically). LaTeX must then be run again to incorporate the citations and the bibliography into the main document.

An `\addcontentsline` will also be needed to create a line for the bibliography in the table of contents, as described in Section 2.5.3.

6.5 Citing Bibliography Entries

As previously mentioned, to produce a citation of a bibliography entry, use the `\cite` command with the the first item in the `.bib` file entry as argument. Thus, given the `.bib` file entry

```
@book{LaTeXCompanion,
  author        =
    "Michael Goossens and Frank Mittelbach and Alexander Samarin",
  title         = "The \LaTeX\ Companion",
  publisher     = "Addison-Wesley",
  address       = "Reading, MA",
  year          = "1994"
};
```

the command `\cite{LaTeXCompanion}` produces "[6]" in the main document.

6.5.1 The "Preferred" Citation Style

In the "preferred" citation style in the ECE "Reference guide: IEEE style" [8], each reference in a group is cited in its own pair of square brackets, and a range of citations is indicated by an en dash *between* the brackets for the first and last citations. To make this style, use `\cite` for the first and last reference, set the en dash, and use `\nocite` for the references in between:

```
\cite{IEEEexample:articleetal}%
--\nocite{IEEEexample:article_typical}%
\cite{IEEEexample:conf_typical}%
```

produces "[11]–[13]" (these references are included as examples in the file `IEEE_ECE.bib`). Only [11] and [13] are shown, but [12] also gets numbered and added to the bibliography, so it can be cited later, as in this sentence.

6.5.2 The "Acceptable" Citation Style

The "acceptable" style in the ECE "Reference guide: IEEE style" [8] differs in two respects from the "preferred" style:

- References cited together are grouped within a single pair of brackets.
- The en dash in a range of citations occurs *within* the pair of brackets, between the first and last citations.

To produce the first effect, put all the citations in the group into the argument of a single `\cite` command, separated by commas:

```
\cite{LaTeX, LaTeXCompanion}.
```

produces "[5, 6]."

To collapse more than two consecutive citations into a range within one pair of brackets and separated by an en dash, you must use the `cite` package by setting

```
\usepackage{cite}
```

in the preamble of your thesis. Then

```
\cite{Impatient, TeXbook, ctan}
```

will produce "[1–3]" instead of "[1, 2, 3]" as would happen without the use of the `cite` package.

6.5.3 Comparison of the Citation Styles

Notice in the last example in Section 6.5.2 that the `cite` package inserts extra space in front of the left bracket. Therefore, if you use the `cite` package, it is no longer possible to produce the "preferred" style:

```
\cite{Impatient}--\cite{LaTeXCompanion}.
```

now produces a defective result, "[1]– [6]."

The "acceptable" style has one advantage over the "preferred" style. Suppose you cite a group of three references that have not been previously cited.

- If you are using the "acceptable" style (`\cite{refA,refB,refC,refD}`), the `cite` package will automatically collapse the group and set the en dash: [1–4]. Now suppose you are revising your thesis and find it necessary to add a citation to the second reference in the group somewhere before you cite the group: `\cite{refB}`... `\cite{refA,refB,refC,refD}`. The `cite` package will handle the change automatically, producing [1]... [1–4], and even correcting for the references now being no longer in citation order in the group.

- If you are using the "preferred" style, however (that is, the command `\cite{refA}--\nocite{refB}\nocite{refC}\cite{refD}`), then your citations [1]–[4] will become [1]... [2]–[4] when you add an earlier citation for `refB`. Nothing will seem wrong about the group citation (so the Publications Office will not mark a correction), but the reader will no longer be aware that you meant to cite `refB`, now numbered [1], in the group. You will then have to correct the group citation by hand: `\cite{refB}--\nocite{refA}\nocite{refC}\cite{refD}` to produce [1]–[4] as you intended.

CHAPTER 7

CONCLUSIONS

We hope that this introduction has been helpful to you in using LaTeX to produce your thesis. Much more information can be found in [1], [5], and [6]. If you encounter difficulty with the operation of any of the features of the ECE thesis template described here, please contact the ECE Publications Office. We will do our best to fix any bugs that are found in the template (but no guarantees!) or inaccuracies found in this document; bringing them to our attention will benefit future students as well. Of course, we cannot answer every general query about TeX and LaTeX features not specific for the UIUC ECE thesis template. We do keep a small, noncirculating library of TeX and LaTeX reference works in our office, and you are welcome to visit and use them.

REFERENCES

[1] P. W. Abrahams, K. Berry, and K. A. Hargreaves, *TEX for the Impatient*. PDF of book originally published in 1990 by Addison-Wesley, who graciously released the copyright back to the authors, 2003. [Online]. Available: www.ctan.org/tex-archive/info/impatient/book.pdf

[2] D. E. Knuth, *The TEXbook*. Boston, MA: Addison-Wesley, 1986.

[3] The Comprehensive TEX Archive Network (CTAN). [Online]. Available: http://www.ctan.org/

[4] D. E. Knuth, *The METAFONTbook*. Boston, MA: Addison-Wesley, 1986.

[5] L. Lamport, *LATEX: A Document Preparation System: User's Guide and Reference Manual*. Reading, MA: Addison-Wesley, 1994.

[6] M. Goossens, F. Mittelbach, and A. Samarin, *The LATEX Companion*. Reading, MA: Addison-Wesley, 1994.

[7] UIUC Graduate College Thesis Office Web Site. [Online]. Available: http://www.grad.illinois.edu/thesis-dissertation/

[8] "Reference guide: IEEE style," ECE Publications Office, University of Illinois at Urbana-Champaign, Oct. 2009.

[9] *The Chicago Manual of Style*, 14th ed. Chicago and London: The University of Chicago Press, 1993.

[10] The Unicode Consortium, *The Unicode Standard, Version 5.0*. Indianapolis, IN: Addison-Wesley Professional (imprint of Pearson Education), 2006.

[11] C. Holmes et al., "150 GHz of thermo-optic tuning in direct UV written silica-on-silicon planar Bragg grating," *Electron. Lett.*, vol. 45, no. 18, pp. 954–956, 2009.

[12] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," vol. 20, pp. 569–571, Nov. 1999.

[13] R. K. Gupta and S. D. Senturia, "Pull-in time dynamics as a measure of absolute pressure," in *Proc. IEEE International Workshop on Microelectromechanical Systems (MEMS'97)*, Nagoya, Japan, Jan. 1997, pp. 290–294.

AUTHOR'S BIOGRAPHY

In addition to checking the format of every ECE thesis and dissertation for compliance with Graduate College requirements, the ECE Publications Office copyedits theses and dissertations for consistency, clarity, and readability as well as correct spelling, grammar, punctuation, and syntax. Publications Office personnel are available to answer questions about editing and writing issues as well as questions about the use of LaTeX and the ECE thesis template, though we are by no means experts in the use of LaTeX. We keep a small, noncirculating library of TeX and LaTeX reference works in our office, and students are welcome to visit and use these. Our print shop can print and bind your thesis when you are finished with it.

Note: As of 3 October 2011, the Graduate College will no longer accept Author's Biography, CV, or Vita in a thesis or dissertation.