

YOUR THESIS TITLE HERE GIVEN IN ALL CAPS, INVERTED PYRAMID FORMAT

BY ****YOUR NAME HERE****
AUGUST, 2010

This document was prepared by ****Your Name Here**** in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering at the University of Maine.

Advisory Committee

Donald M. Hummels, Professor of Electrical and Computer Engineering, Advisor
Fred H. Irons, Castle Professor of Electrical and Computer Engineering
Allison I. Whitney, Lecturer in Electrical and Computer Engineering

Abstract

This is where your abstract goes. Make it short and to the point. Read the thesis guidelines from the Graduate School.

This documents serves as a template for preparation of an M.S. Thesis at the University of Maine. Formatting of this document is handled by \LaTeX . You should be able to produce a beautiful thesis just by changing the text in this document.



© August, 2010 ****Your Name Here****

ACKNOWLEDGMENTS

Optional Section... but usually required if your work was externally supported.

This work has been supported in part by the ARPA HBT/ADC program under a contract administered by the Office of Naval Research Grant N000149311007, and the Army Research Office Grant DAAH04-94-G-0387.

Many people also thank their parents, their advisor/committee, and their friends. It's your call.

Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you.

Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you. Thank you, Thank you, Thank you.

Contents

ACKNOWLEDGMENTS	ii
List of Tables	iv
List of Figures	v
LIST OF ACRONYMS	vi
1 INTRODUCTION	1
1.1 Background	1
1.2 Purpose of the Research	1
1.3 Thesis Organization	1
2 USING THIS TEMPLATE	3
2.1 Required Files	3
2.2 Compiling and Viewing the Document	4
2.3 Modification of Source Files	4
2.3.1 thesis.tex	4
2.3.2 references.bib	4
2.3.3 Private Command Definitions	4
3 L^AT_EX BASICS	5
3.1 Sections and Sub-Sections	5
3.1.1 Example subsection	5
3.1.2 Getting References to Work	6
3.2 Tables and Figures	6
3.3 Lists	7
4 TYPESETTING MATH	8
4.1 Types of Equations	8
4.2 Equations Which Hold Arrays	9
4.3 Other Examples	9

CONTENTS

5	GRAPHICS	10
5.1	latex vs. pdflatex	10
5.2	Including Graphics	11
6	USING BIBTEX	14
6.1	The BibTeX Database: references.bib	14
6.2	Creating the List of References	15
7	ADVANCED FORMATTING	16
7.1	hyperref	16
7.2	Multiple Page Tables	16
7.3	Landscape Tables and Figures	18
7.4	Subfigures	21
7.5	Inserting Source Code (Verbatim Input)	22
7.5.1	Alternatives for Source Code Segments	22
7.6	Acronyms	23
8	FINAL FORMATTING	24
8.1	Figure and Table Placement	24
8.1.1	Deciding Not to Float	25
8.2	Bad Line Breaks	25
8.3	Widows and Orphans	26
8.4	Avoiding Page Breaks	27
8.5	Page Breaks in the Table of Contents, List of Figures, etc.	27
8.6	Adding Table of Contents Entries	28
9	CONCLUSIONS	29
	Bibliography	30
A	SAMPLE APPENDIX FILE	31
A.1	Section of an appendix	31
A.1.1	Subsection of an appendix	31
A.1.2	Subsection of an appendix	31
A.2	Another Section of an appendix	32
B	SAMPLE APPENDIX FILE	33
	BIOGRAPHY OF THE AUTHOR	34

List of Tables

2.1	Files required to produce a thesis. Files which must be modified are indicated.	3
3.1	This is an example table. Various types of \LaTeX dashes and hyphens are illustrated—except for the punctuation dash, which is shown here in the caption.	6
3.2	This table has two columns... The first is 1.5 inches wide and the second is two inches wide.	6
7.1	This is a very long table.	16
7.2	Wide tables may look best in a landscape orientation.	19

List of Figures

3.1	A blank figure example.	7
5.1	Original PostScript or PDF picture	11
5.2	Width scaling to two inches	11
5.3	Height scaling to one inch.	11
5.4	Making image 5 inches by 2 inches	12
5.5	Scaling the figure to occupy 80% of the linewidth.	12
5.6	30 degree rotation of the image.	13
5.7	A bitmap image, stored as a jpg.	13
7.1	Example of a sideways figure.	20
7.2	Example illustrating the use of subfigures.	21
8.1	An inline figure example (usually a bad idea).	25

LIST OF ACRONYMS

CDMA	Code Division Multiple Access	23
CYL	See You Later	23
DUT	Device Under Test	23
H₂O	Water	23
NASA	National Aeronautics and Space Administration	23
OMG	Oh My God	23
T2UL	Talk To You Later	23
VBF	Very Best Friend	23
WiFi	Wireless Fidelity (IEEE 802.11b wireless networking)	23

Chapter 1

INTRODUCTION

This document is a template for the production of a thesis which meets the University of Maine's Thesis Format requirements. The document is formatted using \LaTeX , and most of the text in this example document gives a short introduction to typesetting using \LaTeX . Thesis format requirements are also mentioned where appropriate.

Although a matter of preference, chapters within this template begin with one or two paragraphs introducing the goals of the chapter. The actual work is done in sections and sub-sections within each chapter. Some advisors may wish to omit the introductory paragraphs. The selection of sections within a thesis introduction should be agreed upon with the thesis advisor. The following three sections seem to be typical.

1.1 Background

Discuss work which was done before the work presented in this thesis. Give references to original work, and bring the reader up to speed on the state of the art at the onset of the thesis. This is a "history of the problem."

1.2 Purpose of the Research

Point out the contribution which this thesis tries to achieve. Why is this thesis worth reading?

1.3 Thesis Organization

Set up the reader for the remainder of the thesis. How do you intend to lead him/her through this complicated topic.

Chapter 2 gives an introduction to the mechanics of compiling, viewing, and printing a \LaTeX document in a UNIX environment. Basic \LaTeX commands are introduced

1.3. THESIS ORGANIZATION

in the remaining chapters. Chapter 3 gives a short introduction to the structure of a \LaTeX document, discussing sectioning commands and the use of symbolic references. Examples of Tables and Figures are also given. Chapter 4 gives examples of typesetting mathematical equations. Chapter 5 illustrates the inclusion of Postscript figures into the thesis. Finally, Chapter 6 gives a short introduction to the use of “bibtex” to compile the bibliography.

This document is *not* intended to be a \LaTeX tutorial. There are many excellent books which cover typesetting using \LaTeX . The standard reference is the book by the author of the \LaTeX package [1]. A summary of the \LaTeX 2e extensions for the latest release of \LaTeX may be found in [2].

Chapter 2

USING THIS TEMPLATE

This chapter provides an overview of the mechanics of using the \LaTeX typesetting program with this template to produce a thesis. The commands used to compile and print the document assume a Linux environment using the $\text{t}\epsilon\text{T}\epsilon\text{X}$ distribution.

2.1 Required Files

The files which are required to use the University of Maine \LaTeX formatting tools are summarized in Table 2.1. All of these files should reside in the same directory. In addition to these files, the graphics files `umaine_ece_logo.eps`, `umaine_ece_logo.pdf`, `small_layout.jpg`, `small_layout.eps` and `sample.c` are used for examples, and are needed to compile this document.

File Name	Modify?	File Use
U_Maine_Thesis.cls	No	Meets U. Maine Format Requirements
thesis.tex	Yes	Master file which includes all others
myabstract.tex	Yes	Thesis abstract
acknowledgments.tex	Yes	Thesis Acknowledgments
acronyms.tex	Yes	Acronym definitions
ch1.tex ... conclusions.tex	Yes	Text for various chapters
appA.tex, appB.tex	Yes	Text for the Appendices
references.bib	Yes	Bibtex database
biography.tex	Yes	Author's Biography

Table 2.1: Files required to produce a thesis. Files which must be modified are indicated.

2.2 Compiling and Viewing the Document

The file “thesis.tex” is the master file which contains the \LaTeX source to include the other files as needed. You’ll need to modify this file to match the number of chapters in your thesis, and to select which files to compile on a given run. To compile the \LaTeX source code into a pdf document, type

```
pdflatex thesis.tex
```

This should create the PDF file “thesis.pdf” which can be viewed using “acroread” or your favorite pdf viewer.

2.3 Modification of Source Files

Modification of most of the source files is self explanatory. The text in the abstract, acknowledgments, chapters, and biography must be modified for your thesis. In addition to these, the following files need some special attention.

2.3.1 thesis.tex

This is the master file which must be modified for each thesis. Enter the degree, author, and committee information. Modify the file to match the number of chapters in your thesis.

You will also need to modify this file if you intend to use any other \LaTeX formatting packages.

2.3.2 references.bib

This is the bibliography database file, which will be used by bibtex to produce the bibliography in the correct format. Each citation that you make should have an entry in this file. Every entry in this file begins with a symbolic name, which is used in the citation. For example, check out this citation [3]. Chapter 6 discusses how to use this database to create the bibliography.

2.3.3 Private Command Definitions

Use the “Private Definitions” section of the `thesis.tex` file to create custom commands for things that you just get tired of typing, or things that may change as you are writing your thesis. For example, the file now defines the command `\X` to represent `\ensuremath{\vec{x}}`. Now, to represent the vector \vec{x} , only the string `\X` needs to be typed. Even better, when your thesis advisor decides that he/she wants vectors typeset in bold-face (after you’ve completed three chapters full of equations, you only need to change one line (the definition of `\X`).

Chapter 3

L^AT_EX BASICS

This chapter provides an introduction into how to structure each chapter of your thesis. It also illustrates some of the basic formatting tools (lists, numbered lists, tables, and figures).

3.1 Sections and Sub-Sections

The `\section`, `\subsection`, and `\subsubsection` commands are used to create numbered divisions within a thesis. L^AT_EX will create the numbers for you, so references to section numbers are handled through the use of the `\label{}` and `\ref{}` commands. For example, This is Chapter 3, and the previous chapter is Chapter 2. Note that if additional chapters were added, the symbolic references would still point to the desired location, even though the chapter numbers would change. The `\pageref{}` command can be used to get the page number of a label, so you can say things like “Check out Table 3.1 on page 6.”

3.1.1 Example subsection

Here’s an example of a subsection.

Example subsubsection

This is a subsubsection within Subsection 3.1.1.

Another subsubsection

Another subsubsection within Subsection 3.1.1. Referring to the previous subsubsection gives Section 3.1.1. If you’re using the UMaine thesis class, the reference should look ok. The default report classes don’t number subsubsections, though, and in that case the reference will just point to the subsection (the same as 3.1.1).

3.2. TABLES AND FIGURES

Name	L ^A T _E X Command	Use	Example
hyphen	–	Dash between words	inter-word
en-dash	--	Dash between numbers	Pages 3–8
em-dash	---	Punctuation Dash	(see caption)
minus	\$- $\$$	Math mode subtract	0, 1, and –1

Table 3.1: This is an example table. Various types of L^AT_EX dashes and hyphens are illustrated—except for the punctuation dash, which is shown here in the caption.

Example	This is a two-column examples in which the paragraph column description is used
You can have lots of text in one of these columns, without getting a super-wide table.	This is some text
Equations still work too.	$y(t) = \cos(2\pi ft)$

Table 3.2: This table has two columns... The first is 1.5 inches wide and the second is two inches wide.

3.1.2 Getting References to Work

L^AT_EX pulls the section number information out of “.aux” files which were created on the *previous* run of L^AT_EX. Usually, if section labels have changed, it is necessary to re-compile the file *twice* to get the correct numbers into the symbolic references.

If the references don't look right, try recompiling!

Keeping track of all of the labels can get cumbersome. The process can be simplified by uncommenting the “\usepackage{showkeys}” line in the `thesis.tex` file. Now, when you compile your document, the typeset output will show the symbolic label associated with each `\label`, `\cite`, `\ref`, or `\pageref` command. Try it on this document, and you'll see!

3.2 Tables and Figures

Tables and Figures are “floating” bodies, which means L^AT_EX will try to put them anywhere that it thinks will work. Don't fight this until you have to, when your document is virtually in its final form. An example of a table was given in Chapter 2. Another example is given in Table 3.1. Notice that the `\label{ }` is contained within the table caption.

In this example, L^AT_EX adjusts the column width of the tables to fit the contents. You can also explicitly set the width of a table column. Check out Table 3.2.

Figures work the same way. Figure 3.1 is a figure with three inches of blank vertical space. This could be used if a an item is to be taped into place for the final document. Examples of including graphics in figures are given in Chapter 5.

Figure 3.1: A blank figure example.

3.3 Lists

Lists may be numbered or not. Here's an example of a numbered list with references to the numbered items. Item number 1 is referenced here, and within the list.

1. Here is the first item.
2. And here is the second.
3. This can go on for a long time. This item is longer than item 1. Notice how the numbers in the list can be referenced.

We don't have to number the items. Here's an example of the above list without item numbers.

- Here is the first item.
- And here is the second
- This can go on for a long time. This item is longer than item. We can't put labels on entries in a non-numbered list.

Chapter 4

TYPESETTING MATH

Using \LaTeX to typeset equation is easy, but takes some time to learn. This chapter gives some examples of different types of equations.

4.1 Types of Equations

Equations which appear within the text are delimited by dollar signs, so that $\$y(t) = \cos(2\pi ft)\$$ becomes $y(t) = \cos(2\pi ft)$. Equations with numbers use the `equation` environment, as in

$$y(t) = \cos(2\pi ft). \quad (4.1)$$

Note the punctuation at the end of equation (4.1), since it completes the above sentence. Note also that `\label{}` and `\ref{}` also work for equations.

Arrays of equations may be created using the `eqnarray` environment, or (usually better) the `amsmath` defined `align` environment.

$$z(t) = x(t) + y(t) \quad (4.2)$$

$$= x(t) + \cos(2\pi ft). \quad (4.3)$$

You can avoid numbering each element of an equation array by using the `\nonumber` keyword. For example

$$\begin{aligned} z(t) &= x(t) + y(t) \\ &= x(t) + \cos(2\pi ft). \end{aligned} \quad (4.4)$$

To avoid numbering any of the equations, use the `align*` environment as follows:

$$\begin{aligned} z(t) &= x(t) + y(t) \\ &= x(t) + \cos(2\pi ft). \end{aligned}$$

This is also the method of getting a single un-numbered equation.

$$y(t) = \cos(2\pi ft).$$

4.2 Equations Which Hold Arrays

Many equations hold arrays of numbers or other equations. You can use the `array` environment to typeset these. It works about the same way as the `tabular` environment used earlier for tables.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & x^2 \end{bmatrix}. \quad (4.5)$$

A cleaner construct of the above uses the `amsmath` defined `bmatrix` environment:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & x^2 \end{bmatrix} \quad (4.6)$$

Check the `amsmath` documentation for many other useful constructs.

4.3 Other Examples

Here's an example with several cases:

$$z(t) = \begin{cases} 0 & \text{if } t < 0, \\ \cos(2\pi ft) & \text{whenever } 0 \leq t < 1, \\ 1/(1+t^2) & \text{for } t \geq 1 \end{cases} \quad (4.7)$$

Here's an equation with an integral:

$$x(L) = \int_{-\infty}^{2L} y(t) \cos(t) dt. \quad (4.8)$$

Summations work the same way. Use the `\left` and `\right` modifiers on delimiters to make them scale to the size of the enclosed contents.

$$g(t) = \left(\sum_{i=0}^{\infty} c_i e^{j2\pi f_0 t} \right)^2 \quad (4.9)$$

Finally, here's a fraction:

$$H(\omega) = \frac{1}{1+j\omega}. \quad (4.10)$$

Chapter 5

GRAPHICS

This chapter gives a summary of commands used to insert and manipulate computer generated graphics. The discussion is based on the “graphicx” package.

5.1 latex vs. pdflatex

Depending on whether you are using the “latex” or “pdflatex” command to compile your document, different graphics formats are supported.

Traditionally, “latex” supports Encapsulated Postscript (.eps) files. Most techniques to include other graphics formats involve first translating the desired file into an EPS format. The linux “convert” program can be used to interpret many common graphics formats. For example, to create an EPS version of a JPEG image called “small_layout.jpg”, use

```
convert small_layout.jpg small_layout.eps
```

Unfortunately, the newer “pdflatex” supports a number of graphics formats, but *not* EPS (it does support encapsulated PDF, and JPEG). Encapsulated PostScript files must be translated into PDF format. One method is to use the “epstopdf” command, as in

```
epstopdf umaine_ece_logo.eps
```

This will create the file “umaine_ece_logo.pdf” which can be directly included when compiling using “pdflatex”.

I tend to use both the “latex” and “pdflatex” commands, and so I prefer to have both an EPS file and a pdflatex-compatible format file (jpg or pdf) available. When I include a graphics file in my document, I *do not specify the file extension*. Both “latex” and “pdflatex” are smart enough to look for graphics formats that they understand, and will read the appropriate file. This way, I don’t have to modify my L^AT_EX source file when I want to create a PostScript or PDF output. For example, to include the above “small_layout.jpg” image, I would use



Figure 5.1: Original PostScript or PDF picture



Figure 5.2: Width scaling to two inches

```
\includgraphics{small_layout}
```

Then “pdflatex” would find and use the file “small.layout.jpg”, and “latex” would find and use the file “small.layout.eps”, which was created by the above translation.

5.2 Including Graphics

Here are a few examples. Figure 5.1 just includes a graphic created using oodraw. I exported the figure from oodraw into an EPS format, and then used the `epstopdf` command to create a PDF format graphic to support `pdflatex`. This creates a nice vector graphic which can be scaled to any size. (oodraw will export directly to a PDF document, but it insists on setting the size of the export to the page-size of the document. You can set the page size in oodraw to nicely enclose the graphic, and then export directly to PDF. For me, it’s easier to just select the graphic, export it to EPS, and then use `epstopdf` to get the PDF file.)

You can scale either the width or height of the picture as illustrated in Figures 5.2 and 5.3. Scaling the width and height creates a stretched image, as shown in Figure 5.4. To avoid distorting the graphic, and to set the width to scale nicely if the margins of the document are changed, it’s a good idea to scale only width, and to base the scale factor on the current linewidth. This is done in figure 5.5.



Figure 5.3: Height scaling to one inch.

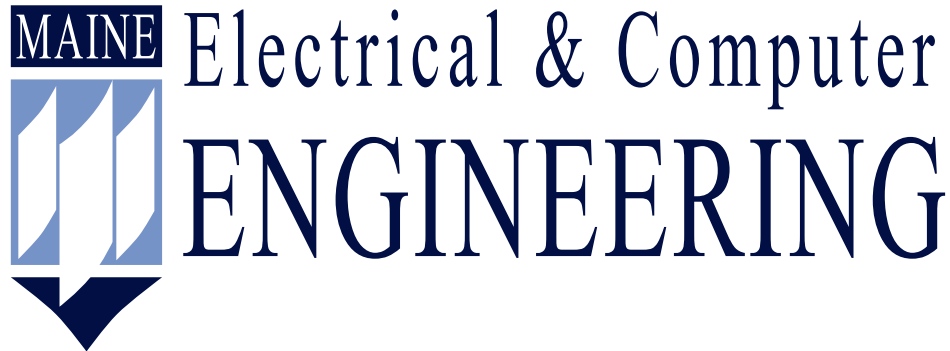


Figure 5.4: Making image 5 inches by 2 inches



Figure 5.5: Scaling the figure to occupy 80% of the linewidth.

The image may also be rotated by specifying a rotation angle in degrees. Check out Figure 5.6.

Bitmap images can also be included. Figure 5.7 shows a JPEG image of a layout. The .jpg file is directly included by “pdflatex”. To support “latex”, the file was converted to an EPS file using the “convert” program as given in Section 5.1.

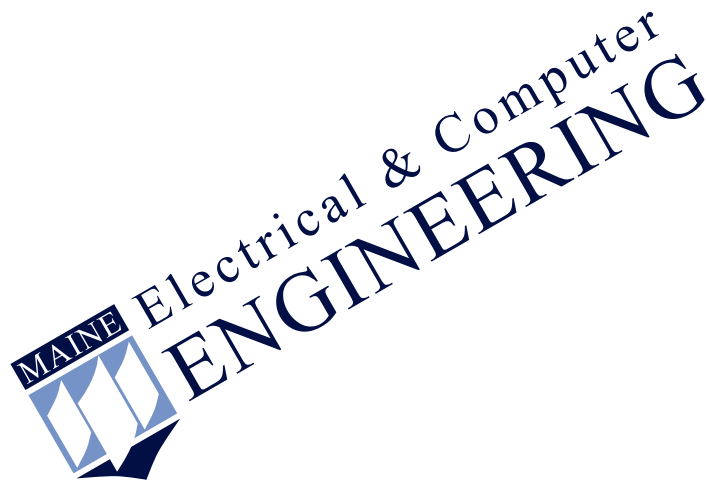


Figure 5.6: 30 degree rotation of the image.

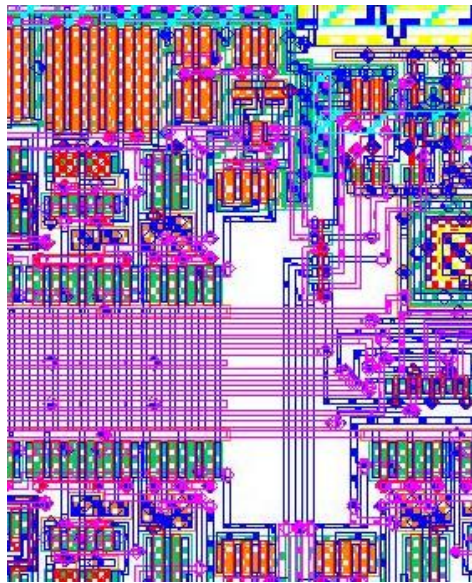


Figure 5.7: A bitmap image, stored as a jpg.

Chapter 6

USING BIBTEX

It is possible to typeset the references for your thesis by hand, and if you do not have many references, this may in fact be the easiest path to follow. The procedure is described in [1]. For more complicated documents, BibTeX is a powerful tool for typesetting the bibliography of your thesis. For example, note the differences between citations of articles [4], books [3], tech reports [5], manuals [6], and conference proceedings [7]. Each type of citation has its own rules governing what font is used for titles, journal names, etc. BibTeX handles these details for you, and organizes your citations into an accurate bibliography. This chapter (hopefully) gives enough information for you to take advantage of this tool. A more complete description of BibTeX is given in [2].

BibTeX's database language is a bit cryptic, so you may wish to use a custom browser/editor to create your bibliography file. Possible tools include `gbib`, `BibDB`, `barracuda`, and `pybliographer`.

6.1 The BibTeX Database: references.bib

Each citation that is used in your thesis must have an entry in the BibTeX database file, which is “references.bib” for this thesis template. Enter your own references following the examples given in that file. The first part of each database entry is a symbolic name, which will be used to create the citation. The `\cite` command is used to add a citation to a given database entry. For example, `\cite{Hummels_95_1}` will become “[7]”, since this article has the symbolic name “Hummels.95_1” in the database file. BibTeX figures out which files within the database you are citing, and creates the typeset bibliography for you.

6.2 Creating the List of References

Because of the interaction between L^AT_EX and BibTeX, a more complicated procedure is needed to create the complete bibliography, and get the references right within the L^AT_EX document. The following procedure should work.

1. Compile the L^AT_EX document using the usual command

```
pdflatex thesis.tex
```

This creates files which BibTeX needs to determine what to include in the list of references.

2. Run BibTeX to compile the bibliography.

```
bibtex thesis
```

This creates a file which L^AT_EX will include to produce the bibliography.

3. Compile the L^AT_EX document again using the usual command

```
pdflatex thesis.tex
```

This line will include the BibTeX bibliography in the resulting document. However, since citation numbers are obtained from the previous compilation of L^AT_EX, the citation numbers will probably not be correct.

4. Recompile the file to get the citation numbers correct.

```
pdflatex thesis.tex
```

Chapter 7

ADVANCED FORMATTING

This chapter illustrates the use of specialized \LaTeX macro packages to accomplish some more advanced formatting requirements the show up from time to time. They are supported only by including the appropriate `\usepackage{}` command in the “thesis.tex” file.

7.1 hyperref

The `hyperref` package supports symbolic links within the document, particularly for the PDF output file format. In addition, hyperlinks to www sites may be included in the document. For example, you can obtain documentation for the `graphicx` package by clicking [HERE](#).

7.2 Multiple Page Tables

For lots of data, you may need a table that spans several pages. Table 7.1 shows an example of such a table. Note that the table is typeset single-spaced, which looks better and has been accepted by the Graduate School in past years.

Table 7.1: This is a very long table.

FIRST	SECOND
First line	This is the actual table data. I'll enter enough to fill a few pages.
Second line	This could take a while.
Third line	I'm getting tired of typing.
Other lines	More of the same.
Other lines	More of the same.

(Continued on next page)

7.3. LANDSCAPE TABLES AND FIGURES

Context	Length	Breadth/ Diameter	Depth	Profile	Pottery	Flint	Animal Bones	Stone	Other	C14 Dates
Grooved Ware										
784	—	0.9m	0.18m	Sloping U	P1	×46	×8	—	×2 bone	2150± 100 BC
785	—	1.00m	0.12	Sloping U	P2-4	×23	×21	Hammerstone	—	—
962	—	1.37m	0.20m	Sloping U	P5-6	×48	×57*	—	—	1990 ± 80 BC
983	0.83m	0.73m	0.25m	Stepped U	—	×18	×8	—	Fired clay	—
Beaker										
552	—	0.68m	0.12m	Saucer	P7-14	—	—	—	—	—
790	—	0.60m	0.25m	U	P15	×12	—	Quartzite-lump	—	—
794	2.89m	0.75m	0.25m	Irreg.	P16	×3	—	—	—	—

Table 7.2: Wide tables may look best in a landscape orientation.



Figure 7.1: Example of a sideways figure.



(a) The small version.



(b) The large version.

Figure 7.2: Example illustrating the use of subfigures.

7.4 Subfigures

The `subfigure` package may be used to create and manipulate figures with multiple parts. Figure 7.2 shows an example, in which Figure 7.2(b) is a magnified version of 7.2(a).

7.5 Inserting Source Code (Verbatim Input)

In some cases, program source code or variable names show up in a document. I'll usually use the `\verb` command to give a variable within the text of a file, as in the variable "complex_sig".

The `verbatim` environment can be used to include a short segment of code directly in the document. For example

```
for (i=0; i<N; i++) {
    // Do some work here
}
```

This all works fine for short code segments, but there are things to watch out for. The text you type into the `verbatim` environment will show up exactly as you type it using a fixed width font. Tab characters will not have the expected look, so you'll have to replace them with spaces. For larger programs, a better scheme is to directly import the program file, and specify the tab spacing that was used for the file. The `\verbatiminput` command in the `moreverb` package lets you do this. Here's an example using a tab spacing of eight characters.

```
#include <stdio.h>

int main(void) {
    int i;
    printf("hello, world\n");
    for (i=0; i<10; i++) {
        printf("  I said HELLO!\n");
    }
    return 0;
}
```

7.5.1 Alternatives for Source Code Segments

Verbatim commands are easy, but they tend to cause trouble as the document is re-sized or margins are adjusted. For a few code segments, you might consider "prettyprinting" the source code into an encapsulated postscript form, and just including the result as a graphic. This provides a nice syntax highlighting, and allows you to make sure that the code-segment will fit in your margins.

Another option is to use the `listings` package. For any document with lots of code, it's worth checking out.

7.6 Acronyms

There are several ways to handle acronyms in \LaTeX . In this document, I used the `acronym` package and assembled the list of acronyms by hand.

I'll guess that some of the boys at National Aeronautics and Space Administration (**NASA**) have heard about the package via their Wireless Fidelity (IEEE 802.11b wireless networking) (**WiFi**) or Code Division Multiple Access (**CDMA**) data links. The key idea is that the first time an acronym is used, it gets defined. After that first definition, the short form is used. Oh My God (**OMG**)! It'll even do this while you're drinking a glass of Water (**H₂O**). Your Very Best Friend (**VBF**) might tell you See You Later (**CYL**), or perhaps Talk To You Later (**T2UL**)...

I'll guess that some of the boys at **NASA** have heard about the package via their **WiFi** or **CDMA** data links. The key idea is that the first time an acronym is used, it gets defined. After that first definition, the short form is used. **OMG**! It'll even do this while you're drinking a glass of **H₂O**. Your **VBF** might tell you **CYL**, or perhaps **T2UL**...

There are lots of twists to get the acronyms to look right. You can explicitly use the short form as in "**DUT**", the long form as in "Device Under Test", or the full form to redefine as in "Device Under Test (**DUT**)". Plural forms are generated as well, as in "**DUTs**" or "Devices Under Test". To "forget" about previously used acronyms so that they are redefined on their next use, use the `\acresetall` command. This is probably a good idea after the thesis front-matter (abstract, title page, etc.) if any acronyms are used there. Be careful placing the acronym in any heading. The "first use" of the acronym might end up being in the table of contents (probably not what you want), and `pdflatex` doesn't seem to deal with it very well. Read the `acronym` package documentation (or just ask one of your new **VBFs**).

Chapter 8

FINAL FORMATTING

This latex template will only get you close in your quest for thesis format approval. Invariably, you'll need to "tweak" the final version of the thesis in order for it to pass the final test. Put this process off until all other editing is done, and your thesis is being typeset for the last time. Here are a few tricks to deal with the complaints that the format reviewers are bound to put on your plate.

8.1 Figure and Table Placement

\LaTeX will place the figure or table wherever it thinks it belongs. Get over it. Let it do its thing. Don't spend time fighting it until you're down to the final copy of your thesis. Eventually, you'll have to deal with these "floats" that move from where they're defined. You can specify some placement suggestions in the `figure` or `table` environments. But these are only suggestions, and the actual location will "float" back in the document from the location of the environment until \LaTeX thinks it has a good opportunity for placement.

If you're going to do battle, though, you may as well do it right. Here are a few suggestions.

- Check the figure placement parameters for the figure or table. \LaTeX 's default (`tbp`) isn't bad. You might consider adding an "h". But don't omit the "p" parameter. Doing so will tell \LaTeX that if you can't put the figure *here*, you shouldn't put it anywhere. The result will be floats that pile up and are pushed farther and farther back into the document. Putting an explanation mark *before* a placement parameter causes \LaTeX to try "extra hard" to place as requested (e.g., `\begin{figure}{h!tbp}` strongly suggests placement at the top of the page in which the figure is defined.) The ordering of the placement parameters is ignored.
- The `\clearpage` command forces \LaTeX to finish placing any pending figures and to start on a new page.

- Try using the `placeins` package in the document preamble. A package option will prevent any floats from moving beyond a `\section` command (use `\usepackage[section]{placeins}`). Also, the package defines a `\FloatBarrier` command beyond which floats are not allowed.
- The `afterpage` package may help. It defines an “`\afterpage`” command which lets you execute a formatting command after the current text on a page is processed. Inserting “`\afterpage{\clearpage}`” should execute a `\clearpage` after the current page is typeset. This would force the placement of the floating figures and tables, without breaking the flow of the text.
- You can include the package “`float`” in the document preamble. This will define a new placement parameter “`H`”, which tells \LaTeX to place the figure immediately, regardless of how bad it thinks it might look. This is a bad idea, but you’re going to find out about it anyway, so here it is. If you’re taking this step, you may as well not use the `figure` or `table` environments. Read Section 8.1.1.

8.1.1 Deciding Not to Float

By the time you’re trying to force the position of a figure or table (by specifying “`H`” as a placement option), there’s not very much point in using a floating environment to start with. Graphics or tabular data can always be inserted into the document, without using the `figure` or `table` environment. The thesis format class file includes the `ccaption` package, which provides a method of adding a caption to these non-floating entries. To define “fixed caption” commands, use

```
\newfixedcaption{\figcaption}{figure}
\newfixedcaption{\tabcaption}{table}
```

Now, “inline” figures and tables can be introduced, labeled, and included in the list of figures/tables. An example is shown in Figure 8.1 below.



Figure 8.1: An inline figure example (usually a bad idea).

More text can follow the figure as is shown here. More text can follow the figure as is shown here. More text can follow the figure as is shown here.

8.2 Bad Line Breaks

To prevent a line break from happening, use a “nonbreaking space” signified by the `~` character. For example, typing “`Section~\ref{sectiontag}`” instructs \LaTeX never to type the word “Section” on a different line from the section number.

The opposite problem is when \LaTeX can't find a good place to break your text, and pushes things into the margin. You'll always get a few "overfull hbox" warnings when you compile, and they're at least worth checking out to make sure that the margins are not way out of control. In many cases, the errors are small enough that you may get away with ignoring them. If not, you can often make the problem go away with a small change in the wording. Here are a few other options to control the placement of a line break:

- Sometimes, \LaTeX doesn't know how to hyphenate a word, and you can insert a "\-" into the word to suggest a place that a valid hyphen can be inserted. If the word is used throughout the thesis, use the `hyphenation` command in the preamble to teach \LaTeX the valid hyphenation points. For example, if your thesis involves a discussion of sesquipedalian words, you might want to add the following to `thesis.tex`:

```
\hyphenation{ses-qui-pe-da-lian}
```

- You can influence \LaTeX 's willingness to hyphenate words (throughout the document) by modifying the `\righthyphenmin` and `\lefthyphenmin` in the `U_Maine_Thesis.cls` file.
- In final formatting, you may have to manually insert `\linebreak` or `\newline` commands to clean things up. This was done in section 8.1. The `linebreak` command also attempts to expand the broken line to fill the margins, while `\newline` does not.
- Long unbreakable words are often the culprit. In this document, "verbatim" text is often a problem. Verbatim text is never broken (that's the point). If you don't mind if a line break occurs in one of these elements, you should probably just change the font, instead of declaring a block of text as "verbatim". For example, use `"long_variable"` instead of `"long_variable"`.

8.3 Widows and Orphans

The thesis format reviewers are awfully good at finding widows and orphans, in which a paragraph is started at the bottom of a page, or ended at the start of a new page. The `\clubpenalty` values in the class file are designed to strongly discourage these, but you'll still see a few. You may re-word sections of your text to avoid them, or consider the following tricks:

- You can include the `needspace` package in the document preamble, and use something like `\needspace{5\baselineskip}` to tell \LaTeX that you need at least five lines on the page for what follows. If it's not available, a new page should be started. To be more insistent, try using `\Needspace*{xxx}` instead of `\needspace{xxx}`.

- To sneak in a page that's one line longer than normal, insert `\enlargethispage{\baselineskip}` into the document. To shorten a page, try `\enlargethispage{-\baselineskip}`. The command may have to land between paragraphs (I haven't checked).
- Inserting `\pagebreak` into text instructs latex to use a page break at the end of the current line of text that it's typesetting.
- Similarly, inserting `\nopagebreak` into the text instructs \LaTeX not to allow a page break at the end of the current line.
- Sometimes you can sneak a little extra space in between paragraphs by using something like `\vspace{0.05in}` in the above paragraph. (This would add 0.05 inches of extra space at the end of the paragraph in which it is inserted.)
- You may be able to fit a little more or less on a page without violating the margins by temporarily modifying the inter-line spacing on a page. A "stretch" factor controls the spacing (set in the `U_Maine_Thesis.cls` file). For the double-spaced thesis, the default value is `\setstretch{1.655}`. So to change a segment of the text to a slightly tighter layout, you might insert `\setstretch{1.5}` at the beginning of the segment, are reset to the normal double-spacing at the end of the segment using `\doublespaceing` (equivalent to `\setstretch{1.655}`). Conversely, you could expand the layout slightly by using a stretch factor slightly larger than 1.655.

8.4 Avoiding Page Breaks

Sometimes, block of text should logically be kept together. \LaTeX 's `samepage` environment should keep a block together, but it will usually fail you just where you need it. For example, `samepage` fails on list environments that suggest their own page breaks between items. If it *does* work, it'll often result in text running off the bottom of the page. If you really need a block of text held together, enclose the text in a box using either `\parbox{}` or `minipage`. You can see an example in Figure 8.1, where I never want the graphics in the figure separated from the caption. Then use the tricks from Section 8.3 to make sure that the box lands in a reasonable place.

8.5 Page Breaks in the Table of Contents, List of Figures, etc.

You may get undesired page breaks in the latex-generated lists in the front-matter of the thesis. A chapter entry may be introduced on one page, while all of its subsections are on another. Or you may have a long multi-line title which gives a page break between the lines. After your document is stable, you can use `\addtocontents` at appropriate places in the body of the document to adjust the page breaking.

- `\addtocontents{toc}{\protect\newpage}` forces a page break.

8.6. ADDING TABLE OF CONTENTS ENTRIES

- `\addtocontents{toc}{\protect\enlargethispage{-2\baselineskip}}` makes a single page of the table of contents two lines shorter.
- `\addtocontents{toc}{\protect\needspace{2\baselineskip}}` specifies to start a new page if there's not a vertical space of two baselines left on the current page. To use the `\needspace` macro, you'll need to add the `needspace` package to the document preamble (using `\usepackage`).

In all cases, you can substitute `lot` or `lof` for `toc` to change the list of tables or list of figures.

8.6 Adding Table of Contents Entries

Occasionally, a student is asked to make a non-standard modification to the table of contents or list of figures/tables. The `\addcontentsline` command is used throughout this template to create entries that point to a specific page of the document. The `addtocontents` command is used to introduce arbitrary commands into the processing of the table of content as illustrated in Section 8.5. (The `\protect` command is used for these entries to instruct \LaTeX to actually write the command into the `toc/lof/lot` files rather than to immediately execute them.) The same mechanism can be used to create non-standard entries.

For example, one recent software-related thesis included an attached CD containing an electronic version of the source code. The CD was placed in a pouch attached to the back cover of the thesis. To place an entry in the table of contents referring to the CD pouch (rather than a page number), the following was added to the end of the thesis (just before the `\end{document}` in `thesis.tex`). The `\contentsline` command typesets the chapter-level entry. `\addtocontents{toc}` writes this command into the `.toc` file, so that it will be executed the next time the table of contents is typeset. The resulting entry in the table of contents is shown in the box below. Hopefully, its format matches the other chapters in the table of contents.

```
\addtocontents{toc}{\protect \contentsline{chapter}
                    {My Glorious Source Code}{CD Pouch}{}}
```

My Glorious Source Code	CD Pouch
--------------------------------	-----------------

Chapter 9

CONCLUSIONS

Here's where the conclusions go. Sum it all up. Make clear the contributions of the thesis. Give an indication of what else could be done in the future.

Bibliography

- [1] L. Lamport, *TEX User's guide & reference manual*. Reading, Massachusetts: Addison-Wesley, 1986.
- [2] M. Goossens, F. Mittelbach, and A. Samarin, *The TEX Companion*. Reading, Massachusetts: Addison-Wesley, 1994.
- [3] F. H. Irons, *Bogus Book Title*. Reading, Massachusetts: Addison-Wesley, 1965.
- [4] S. Silberman-Hummels, "Bogus journal article," *IEEE Trans. Circuits and Systems*, vol. 23, pp. 44–52, June 1994.
- [5] S. P. Kennedy, "Bogus technical report," tech. rep., Naval Undersea Warfare Center, Providence, Rhode Island, Nov. 1988.
- [6] Z. M. Hummels, *Bogus Manual or User Guide*. University of Maine, Orono, Maine, 1991.
- [7] D. M. Hummels, W. Ahmed, and F. H. Irons, "Measurement of random sample time jitter for ADCs," in *Proceedings of IEEE International Symp. on Circuits and Systems*, (Seattle), pp. 708–711, May 1995.

Appendix **B**

SAMPLE APPENDIX FILE

This is an example of an appendix. Put anything that you want here.

BIOGRAPHY OF THE AUTHOR

This is your biography, written in third person. Only the last line of this dude is dictated by the Graduate School. This is a sample taken from an old thesis.

Shawn Kennedy was born in Presque Isle, Maine on November 6, 1967. He received his high school education from John Bapst Memorial High School in Bangor in 1986.

He entered the University of Maine in 1986 and obtained his Bachelor of Science degree in Electrical Engineering in 1990.

In May 1990, he was enrolled for graduate study in Electrical Engineering at the University of Maine and served as Research Assistant . His current research interests include communications and signal processing. He is a member of IEEE, Tau Beta Pi, Eta Kappa Nu, and Phi Kappa Phi, and his interests include motocross and music. The last line of the bio must read something like this... He is a candidate for the Master of Science degree in Electrical Engineering from the University of Maine in May 1992.