THE FLORIDA STATE UNIVERSITY

COLLEGE OF ARTY SCIENCE

THIS IS MY THESIS TITLE:
BROKEN OVER TWO LINES

By

VÌCTÖR É. IŞMYNE

A Thesis submitted to the
Department of Furtive Studies
in partial fulfillment of the
requirements for the degree of
Master of Science

Degree Awarded:
Summer Semester, 2011

Vìctör É. Işmyne defended this thesis on June 15, 2011.

The members of the supervisory committee were:

Faux Causson Yorverk
Professor Directing Thesis

Verda Boizaar
University Representative

Beauxeau D'Claune
Committee Member

Arlip Zarseeld
Committee Member

Deb O'Nair
Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the thesis has been approved in accordance with the university requirements.

To my parents, who always suspected I'd end up here

ACKNOWLEDGMENTS

Many thanks are due to many people. My major professor didn't know what she was getting herself into when she took me on as a student, and I will always be grateful for her support and guidance. The other members of my committee deserve hazard pay, and this paper would not be the same without their diligence: many thanks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

The following short list of symbols are used throughout the document. The symbols represent quantities that I tried to use consistently.

π	$3.1415926\ldots$
E	mc^2
F	ma
R_e	Mean Radius of the Earth $\approx 6367.65\,\mathrm{km}$
e	Base of Natural Logarithms $\approx 2.71828\ldots$
P	The principal borrowed
N	The number of payments
i	The fractional (periodic) interest rate
P_j	The principal part of payment j
I_j	The interest part of payment j
B	A final balloon payment
x	The regular payment
R	The principal remaining after r payments
r	Some number of payments such that $0 < r < N$
R_j	The principal remaining after j payments
A_j	The total interest paid out after j payments

ABSTRACT

The FSU Thesis Class is a LaTeX document class useful for writing Theses, Dissertations, and Treatises. It has several custom macros and environments which are intended to ease the burden of formatting for writers of these documents so that they may focus more on the research and presentation rather than on the page layout. This sample document is intended to provide a few examples of how most of the class features may be used.

The main source file for this document is `thesis.tex`, and this is where you should start reading. The document's source is spread over several files. Many of the files contain helpful LaTeX comments which are not printed out here. It may be instructive to look at the source files as you read this "output" to see how the document was created.

CHAPTER 1

A DERIVATION OF A FORMULA FOR AMORTIZATION

This chapter contains several examples of the equation environment and equation references. There are also examples of every level of heading, from `\chapter` to `\subparagraph` (though these headings are somewhat artificial). The `amsmath` package is required to process this chapter, and so the `thesis.tex` file in this directory contains the package line `\usepackage{amsmath}`. The text of this file is located in the file `chapter1.tex`. If you're new to LaTeX, you may find it instructive to look at the source text (which contains some extra documentation written as LaTeX comments) while reviewing this printed output.

The text and mathematics in this document are my own work, written in the mid-1980s when I was trying to figure out how long it would take to pay off my credit card debt once I had graduated. (This work is the basis of my on-line amortization calculator at `http://bretwhissel.net/amortization/amortize.html`.) This material became a convenient test bed as I developed the FSU thesis macros.

1.1 Definitions

This is my derivation of the formula for amortization. The goal is to find a payment amount, x, which pays off the loan principal, P, after a specified number of payments, N. Variable definitions are listed in table 1.1.

Table 1.1: The List of Variables

P	The principal borrowed
N	The number of payments
i	The fractional (periodic) interest rate
P_j	The principal part of payment j
I_j	The interest part of payment j
B	A final balloon payment
x	The regular payment

1.1.1 Payment Schedule

Assuming that all payments (excluding an optional final balloon payment) are the same amount, a payment x consists of its interest part and its principal part:

$$x = I_j + P_j \tag{1.1}$$

$$\begin{aligned} I_1 &= iP & P_1 &= x - I_1 \\ I_2 &= i(P - P_1) & P_2 &= x - I_2 \\ I_3 &= i(P - P_1 - P_2) & P_3 &= x - I_3, \text{ etc.} \end{aligned}$$

This schedule states that the payment x includes interest on all of the remaining principal, including that which is part of the current payment. The first payment, therefore, includes an interest payment on the total borrowed, which defines the minimum payment. (If we are to make any progress toward paying off the loan, we must pay more than the amount iP.)

1.2 Recurrence Relation

The P_j's may be rewritten into a recurrence relation:

$$\begin{aligned} P_1 &= x - iP \\ P_2 &= x - i(P - P_1) \\ &= x - i\left[P - (x - iP)\right] \\ &= x - iP + ix - i^2 P \\ &= (x - iP)(1 + i) \\ P_3 &= x - i(P - P_1 - P_2) \\ &= x - i\left[P - (x - iP) - (x - iP + ix - i^2 P)\right] \\ &= x + 2ix + i^2 x - iP - 2i^2 P - i^3 P \\ &= x(1 + i)^2 - iP(1 + i)^2 \\ &= (x - iP)(1 + i)^2 \end{aligned}$$

In general, we will find that

$$P_j = (x - iP)(1 + i)^{j-1}. \tag{1.2}$$

1.3 Balloon Payment

If there is to be a balloon payment, then the final payment will consist of the final principal payment P_f and interest on that principal iP_f so that $B = P_f + iP_f$. Rewriting P_f in terms of B gives $P_f = B/(1 + i)$.

1.4 Finding a Solution

1.4.1 Defining the Equation

Next, we define an equation which uses these ideas:

$$B + Nx = P + \sum_{j=1}^{N} I_j + i\left(\frac{B}{1+i}\right), \tag{1.3}$$

or in English, the sum of all the payments (left side) is equal to the principal borrowed plus all of the interest paid with regular payments plus interest paid on the balloon payment (right side). Note that if there will be no balloon payment ($B = 0$), then the B terms drop out.

1.4.2 Breaking It Down

Now we glue some more pieces together: replace I_j of equation (1.3) using the relationship given by eq. (1.1) and then substitute the recurrence identity of eq. (1.2):

$$\begin{aligned}
B - \frac{iB}{1+i} + Nx &= P + \sum_{j=1}^{N}\left[x - (x - iP)(1+i)^{j-1}\right] \\
B - \frac{iB}{1+i} + Nx &= P + Nx - (x - iP)\sum_{j=1}^{N}(1+i)^{j-1} \\
P - B\left(1 - \frac{i}{1+i}\right) &= (x - iP)\sum_{j=1}^{N}(1+i)^{j-1} \qquad (1.4)
\end{aligned}$$

Initial Solution. Now we can see our way clear to solve for x:

$$x = \frac{P - B\left(1 - \frac{i}{1+i}\right)}{\displaystyle\sum_{j=1}^{N}(1+i)^{j-1}} + iP. \tag{1.5}$$

Finding a Closed Form.

Isolation. The series form of eq. (1.5) can be rewritten without the series after a little transformation. First, we separate the summation and rewrite its limits:

$$x = \left[P - B\left(1 - \frac{i}{1+i}\right)\right]\frac{1}{\displaystyle\sum_{j=0}^{N-1}(1+i)^{j}} + iP. \tag{1.6}$$

Transformation. To simplify the transformation, we can substitute by letting $g = 1+i$ so that the summation looks like $\sum_{j=0}^{N-1} g^j$. Next we multiply the series by $(1-g)/(1-g)$, which causes all but the first and last terms to drop out:

$$\frac{(1-g)\sum_{j=0}^{N-1} g^j}{1-g} = \frac{\sum_{j=0}^{N-1} g^j - \sum_{j=0}^{N-1} g^{j+1}}{1-g} = \frac{1-g^N}{1-g}. \tag{1.7}$$

Result. Since the series is originally in the denominator, we invert the transformed result, and then undo the substitution:

$$x = \left[P - B\left(1 - \frac{i}{1+i}\right)\right]\frac{1-(1+i)}{1-(1+i)^N} + iP. \tag{1.8}$$

Rearranged. Now we can expand and rearrange to taste:

$$x = i\left[\frac{P(1+i)^N}{(1+i)^N - 1} + \frac{B}{(1+i)-(1+i)^{N+1}}\right]. \tag{1.9}$$

Quod erat demonstrandum ("That which was to be shown"), recognized in most mathematical circles as the initials Q.E.D. □

1.5 Application

Two forms. Equations (1.8) and (1.9) solve for the payment amount, but either can be re-arranged to solve for any of the other variables, with the exception of i, the periodic interest rate. To date I have been unable to find an analytic solution for this variable, so the program invokes an iterative method to find successive approximations to the solution.

Computation. To reduce the number of computations, $1+i$ can be stored in single variable, as well as a single calculation of $(1+i)^N$. Then the calculation of $(1+i)^{N+1}$ merely requires multiplying the two previously-calculated values, i.e., $(1+i)^{N+1} = (1+i)^N \cdot (1+i)$.

CHAPTER 2

WORKING WITH GRAPHICS

In this chapter, we'll be dealing with the inclusion and placement of figures and other graphics in your thesis. Usually a figure will be a graphic of some type (e.g., a photograph, line drawing, chart) that resides in a separate file external to your document. You will need to decide on a format for these figures. If you are generating your own graphics using other programs, you may have the option to choose the output format. Otherwise, you may need to use other software to convert the graphic into a compatible format for inclusion in your LaTeX document.

2.1 Raster and Vector Graphics

Graphics for print media are generally in one of two classes: raster images or vector drawings. A JPEG photograph is an example of a raster format, where a matrix of pixels has a defined value. Raster images may be scaled up or down in size, but within limits. Scale such an image too large and it will become "pixelated" or blocky. Reduce the image too much, and finer details will be blurred or lost. If you use raster images in your thesis, they should be generated with a sufficiently high resolution that scaling artifacts will be minimized. For print media, a resolution of 300 to 600 dpi (dots per inch) might be typical. For viewing on screen, a minimum resolution of 100 dpi may be adequate. PNG, JPEG and GIF images are common examples of a raster type. PhotoShop, The Gimp, and other paint-type or photo-manipulation programs typically generate raster images. Scanning documents or other graphics will also generate a raster image of some kind, usually in one of JPEG, TIFF, or PDF formats.

In contrast, vector images will scale better. Vector images are usually line-drawings, text, and many kinds of graphs. Vector images can be scaled better because the pixel values are not defined until after the scaling occurs, so the image can be rendered at the full resolution of the output device, whether it be for print or screen. Also, vector images are often encoded in a much smaller space than a comparably-sized raster image. Adobe Illustrator, CorelDRAW, Inkscape, and Flash are examples of programs that generate vector images. WMF and SVG are also common vector formats, though these may not be included in your documents directly.

In generating figures, if you have a choice, choose a vector format, since it will provide the best scaling flexibility. However, if you must include a photograph or scanned document of some type, then be sure that the raster image has a sufficiently high resolution for the intended publishing medium.

2.2 The graphicx Package

The LaTeX package graphicx allows you to insert external PostScript or other files into your document. There are other figure-inclusion packages that you can use, but graphicx is probably the most common for the task.

This main document file has a `\usepackage[dvips]{graphicx}` line in order to include figures. The option `dvips` indicates that the graphicx package should employ the PostScript driver for the program `dvips` (which converts LaTeX's standard DVI output into PostScript). Using this driver, all your external figures will need to be in Encapsulated PostScript (EPS) form. Other formats will need to be first converted to PostScript. (Both PostScript and PDF formats may contain either raster or vector images, or even both at once.)

An alternative driver to `dvips` is the option `pdftex`, which will require all your figures to be in PDF, JPEG, or PNG format. This will be the preferred driver if you intend to use `pdflatex` to generate your document in PDF form directly, rather than going through an intermediate PostScript form first. All the figures in this document (found in the `figures` directory) are available in both EPS and PDF formats, so you may use either driver. In fact, there's some fancy code in the `thesis.tex` file that chooses the right driver depending on whether you're running `pdflatex` or `latex` to process this document.

2.3 PostScript and Encapsulated PostScript

PostScript is a programming language for putting marks on a page. As such, PostScript files can usually be edited using plain text editors, should that ever be necessary. A file containing Encapsulated PostScript conforms to additional standards, allowing the graphic to be manipulated or included in other PostScript programs more readily. Encapsulated PostScript is the preferred format for use with LaTeX.

As far as LaTeX is concerned, the most important feature of an Encapsulated PostScript file is the `%%BoundingBox`, which is defined as the smallest rectangle that completely encloses the figure. Two sets of x- and y-coordinates describe the bounding box; the first (x, y) pair gives the lower left-hand coordinates of box relative to the bottom left-hand corner of the page, and the second pair of coordinates identify the upper right-hand corner of the bounding box. You normally don't have to worry about this information at all, but if a PostScript figure does not contain `%%BoundingBox` information, or if the `%%BoundingBox` is incorrect, the graphicx package allows you to specify the bounding box coordinates yourself when you insert the figure into your document.

Depending on your distribution of LaTeX, you may have a program already installed which converts EPS files to PDF. One of the simplest converters is the program `epstopdf`, which can be executed as follows:

```
epstopdf myfig.eps
```

A Windows version of this program might pop up a user interface for fine-tuning the figure conversion, while the Unix/Linux version of the program will simply do the conversion, creating `myfig.pdf` without any intervention required.

2.4 Inserting Figures

The basic mechanics of including a PostScript graphic (`dvips` driver) or PDF graphic (`pdftex` driver) are relatively simple:

```
\includegraphics{figurefile}
```

This command will insert the graphic contained in the file `figurefile` at that particular location in the text. The file name may have optional extensions, depending on the driver. For example, the `dvips` driver will recognize the `.ps` or `.eps` file extensions, while the `pdftex` driver will recognize extensions `.pdf` and `.jpg`. LaTeX will leave space for the graphic according to the bounding box information that the file contains.

For inclusion in your thesis or dissertation, a graphic will normally be labeled and given a caption. To do this, you will wrap the `\includegraphics` command in a `figure` environment. Figures will usually "float" from the location where they were defined. By default, LaTeX will try to place figures at the bottom of the current page (if there's sufficient room), then at the top of the next available page. Failing that, LaTeX will place figures onto their own page. The `\caption` command will number the figure automatically. If you need to refer to the figure in your text, include a `\label` command following the `\caption`, and then use the `\ref` and `\pageref` commands to retrieve the figure number and figure's page number, as required.

As a complete example, refer to Figure 2.1. I used the following commands to include the figure graphic:

```
\begin{figure}
\begin{center}
\includegraphics{figures/fleur}
\end{center}
\caption{This is a simple figure, but a complete example.}
\label{ex-complete}
\end{figure}
```

In this case, the graphic file is located in the folder `figures`, and the complete file name is either `fleur.eps` or `fleur.pdf`. If I'm using the `dvips` driver, the `.eps` extension is added automatically; likewise, the `.pdf` extension would be added if I am using the `pdftex` driver. In order to center the graphic horizontally, I enclosed the `\includegraphics` command in a `center` environment. The `\caption` command will handle its text in its own way: a short caption will be horizontally-centered, and a longer caption will be broken into multiple lines as required. At the beginning of the paragraph, I typed `Figure~\ref{ex-complete}` to refer to the figure number. Using this mechanism, I never have to worry about renumbering my figures if I choose to move text or figures around. I can also tell you that Figure 2.1 will be found on page 8 by typing

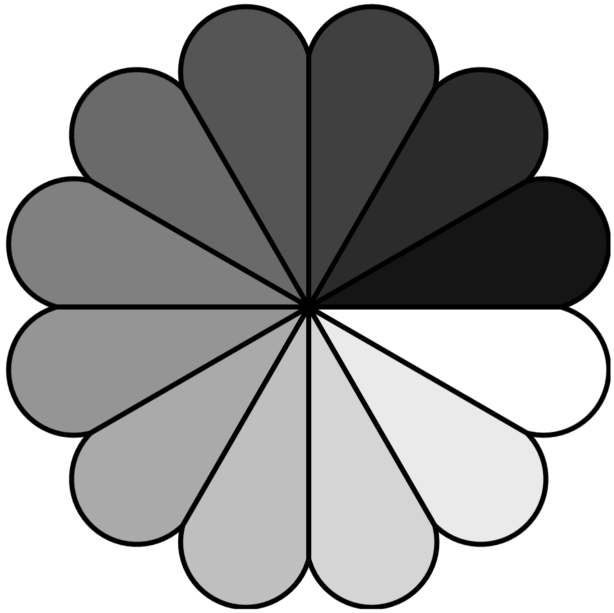

Figure 2.1: This is a simple figure, but a complete example.

```
Figure~\ref{ex-complete} will be found on page~\pageref{ex-complete}
```

The `\includegraphics` command has several optional arguments which will allow you to adjust figure position, bounding box, rotation, scaling, and other elements, without having to edit the figure itself. Figure 2.2 displays an image in its raw state, and then a close-up, rotated image of a segment of the same PostScript graphic. A complete description of the all the options available is best left for the graphicx package manual page, but the transformed image was accomplished with the following `\includegraphics` line:

```
\includegraphics[viewport=.9in 2.1in 1.15in 2.35in,clip,
    scale=10.67,angle=90]{figures/ben}
```

2.5 Musical Examples

If your thesis or dissertation requires you to include musical examples, the fsuthesis class has an environment already set up for you. Each musical example should be in its own PostScript EPS file or PDF file (depending on the graphicx driver you are using). Then, rather than using the `figure` environment, you would use the `musex` environment as demonstrated here to create Example 2.1:

```
\begin{musex}
\begin{center}
\includegraphics{figures/freude}
```

Figure 2.2: Benjamin Franklin, in raw form (left), and close-up and rotated (right). (Notice that the close-up figure displays "pixelation" effects of a raster image scaled too large.)

```
\end{center}
\caption{The bass soloist's opening statement in the 9th Symphony.
  \label{mus:bass-ode}}
\end{musex}
```

As with figures, the caption follows the graphic. However, the caption will be automatically titled with "Example" rather than "Figure". If your document contains more than one musical example, they can be listed in their own table by using the command `\listofmusex` in the front matter section of your document.

Example 2.1: The bass soloist's opening statement in the 9th Symphony.

2.6 Further Information

This chapter intended to provide a brief overview of figure inclusion, along with a few simple examples. If your figure-insertion needs have not yet been addressed, you should look up the document *Using Imported Graphics in LaTeX and pdfLaTeX* by Keith Reckdahl, which is part of the Comprehensive TeX Archive Network (CTAN). This document provides a wealth of detail about figure manipulation, insertion, and placement.

In addition to the inclusion of external figures, LaTeX provides several environments and packages for creating figures mathematically within a LaTeX document itself, usually technical line-drawings. Investigate the picture package and its extensions epic and eepic. Other figures and text manipulations can be carried out with the pstricks package. The program METAPOST can generate precise renderings of mathematical objects and diagrams. And there are many more utilities and packages to assist you.

CHAPTER 3

TABULAR DATA AND TABLES

Most graduate students will come to a place in their career where they must create a table of some kind. Many simple layouts are a breeze with LaTeX. Here's a brief example:

1	a	3^1	3
2	abb	3×3^2	27
33	abbccc	$3^3 \times 3^3 \times 3^3$	19683

and the source text that created it:

```
\begin{tabular}{ r r || c | l }
  1  & a      & $3^1$                            &     3 \\
  2  & abb    & $3\times 3^2$                    &    27 \\ \hline
  33 & abbccc & $3^3 \times 3^3 \times 3^3$ & 19683 \\
\end{tabular}
```

When the `tabular` environment begins, the next required parameter specifies the layout of the table. In this case, the table layout specifies two right-justified columns (`r r`), a double-vertical separator (`||`), a centered column (`c`), a single-vertical separator (`|`), and a left-justified column (`l`). The next rows provide the data for the table, with columns separated by the ampersand (`&`) character. The end of the row is indicated by the double backslash (`\\`). As you can see, the columns may contain text, numeric data, and even some math. A horizontal line may be drawn between rows using the `\hline` command.

For many people, this may be all the information on tables that's required (for now, anyway). But at some point, you may need even more options to create just the right layout. A quick web search for `latex table` will turn up a wealth of usable information, samples, examples, and tutorials.

The `tabular` environment provides the layout mechanism for placing text and data into row and column form. But it's the `table` environment that allows you to automatically number your table and to add a heading (caption). The `table` environment works just like the `figure` environment as far as floating placement is concerned. However, the FSU thesis guidelines state that table captions should appear *before* the table, while `figure` captions appear *after* the figure. The text in Figure 3.1 generates Table 3.1 as an example. (And I used `Table~\ref{sonnets}` in the previous sentence to retrieve the table number.) This demonstrates how one can create paragraphs of text as part of a table by using the `p{5cm}` format specifier. Additional space was inserted after each row by adding a dimension to the linebreak specification, i.e., `\\[5pt]`.

```
\begin{table}
\caption{Shakespeare Sonnets, First Lines, IIX --- XII}
\label{sonnets}
\begin{center}
  \begin{tabular}{r p{5cm} }
    8 & Music to hear, why hear’st thou music sadly? \\[5pt]
    9 & Is it for fear to wet a widow’s eye \\[5pt]
   10 & For shame deny that thou bear’st love to any \\[5pt]
   11 & As fast as thou shalt wane, so fast thou grow’st \\[5pt]
   12 & When I do count the clock that tells the time \\
  \end{tabular}
\end{center}
\end{table}
```

Figure 3.1: LaTeX source that generates Table 3.1 on page 13.

However, just because you put some text or data into a multi-column form, it doesn’t necessarily mean that it’s a table as far as your thesis or dissertation is concerned. If the tabular-form data is part of your text and flows in the order of your presentation, it may not be necessary to set it off as a table. The layout example at the beginning of this chapter is an example of tabular data which is not set off as a table.

Other than the unfortunately confusing similarity in their names, the `tabular` environment and the `table` environment have independent functionality: while the `tabular` environment is often used inside the `table` environment, either environment can be used without the other. And while we’re at it, figures don’t necessarily need to contain graphics. Figure 3.1 is an example of a figure which contains ordinary text, but the text has been wrapped within a `figure` environment so that it can be allowed to float outside the main flow of text.

If you have a particularly wide table, you may want to turn the table sideways on the page. To do this, you should place a `\usepackage{rotating}` in the document preamble. When it is time to insert the rotated table, type `\begin{sidewaystable}` instead of `\begin{table}`. This also works for figures, by the way, so instead of `\begin{figure}`, you may use `\begin{sidewaysfigure}` for diagrams and images that you want rotated. Sideways figures and tables will always be floated to their own page.

Table 3.1: Shakespeare Sonnets, First Lines, IIX — XII

8	Music to hear, why hear'st thou music sadly?
9	Is it for fear to wet a widow's eye
10	For shame deny that thou bear'st love to any
11	As fast as thou shalt wane, so fast thou grow'st
12	When I do count the clock that tells the time

CHAPTER 4

CITATIONS AND REFERENCES

4.1 Using the `references` Environment

Let's start simple. Assuming we don't have many citations, we'll create a references section manually using the `\begin{references}` environment provided by the fsuthesis class. If your discipline has a style guide for presenting references, you can use that to create your own entries. Here are some example entries with explicit formatting specified:

```
\begin{references}
Picaut, J., F. Masia, and Y. du Penhoat, 1997: An advective-reflective
conceptual model for the oscillatory nature of the ENSO.
\textit{Science}, \textbf{277}, 663--666.

Yasunari, T., 1990: Impact of Indian monsoon on the coupled
atmosphere/ocean system in the tropical Pacific.
\textit{Meteor. Atmos. Phys.}, \textbf{44}, 19--41.
\end{references}
```

Note that I had to specify italics and bold-facing myself, according to the style guide that I'm using. (If you don't have a discipline-specific style guide, see [1] or [2] for lots of bibliographic examples.) The `references` environment provides exdented entries, spacing, and a heading, but the rest of the formatting is up to you. Likewise, citations of these references within your document must be formatted manually. Once processed, the reference entries above look like the following:

> Picaut, J., F. Masia, and Y. du Penhoat, 1997: An advective-reflective conceptual model for the oscillatory nature of the ENSO. *Science*, **277**, 663–666.
>
> Yasunari, T., 1990: Impact of Indian monsoon on the coupled atmosphere/ocean system in the tropical Pacific. *Meteor. Atmos. Phys.*, **44**, 19–41.

4.2 Citations and BibTeX

If your thesis or dissertation does not have many citations or references, this may be all you need. However, if you have more than a handful of citations, you owe it to yourself to invest a little more energy into learning about the powerful, time-saving features of BibTeX. To use BibTeX, bibliographic entries are added to an external file with tags identifying elements of the entry, such as authors, titles, journals, etc. You may then cite a reference in your document using its unique key. Many disciplines have developed large BibTeX databases already, so if you're lucky, you only need download a pre-built file ready to go. You can always add a few more references if those entries don't already exist in the file you download.

For this sample document, I have created a small BibTeX bibliography database in a file called `myrefs.bib` which is excerpted from a larger collection of pre-generated TeX-related entries I downloaded from the web. If your discipline does not already distribute BibTeX databases publicly, it's just a bit more typing to create your own BibTeX file. Another advantage of the BibTeX approach is that you can continue to add to this database throughout your professional career, creating entries as you read books and journal articles you may want to reference in the future. I encourage you to refer to the standard LaTeX references (e.g., [4] and [3]) or to search the web for further information on creating your own BibTeX database.

Once you've created or downloaded a BibTeX database, you may cite a document using its unique *key* as an argument to the LaTeX `\cite` macro. For example, in the previous paragraph, I used the commands `\cite{Lamport:1994:LDP}` and `\cite{Kopka:2004:GLT}`, where `Lamport:1994:LDP` and `Kopka:2004:GLT` are the keys for their respective documents. By making these citations, the bibliographic entries for these documents will be pulled from my BibTeX file and added to the bibliography automatically. I can also add entries to the bibliography without having citations in my text by using the `\nocite{key}` command with the desired *key*. If I want every entry in my BibTeX file inserted into my bibliography, then I can issue the command `\nocite{*}` as a shortcut. (You don't have to worry about citations being redundant: bibliographic entries will be included only once no matter how many times they may be `\cite`d or `\nocite`d in your document.)

This document uses the default citation and bibliography formatting supplied by LaTeX, but if your discipline publishes BibTeX files for common use, it may also have created its own BibTeX formatting style which you may want to use. One common alternative is to use the natbib package activated by including `\usepackage{natbib}` in the document preamble. This package will allow you to make author–year citations in your document automatically (e.g., "Fargunkle et al., (2001)") using the same BibTeX database file. See [3] (or search the web) for more information on natbib.

APPENDIX A

NONSENSE

This appendix is here merely to demonstrate how appendices may be included and formatted in your document. Look through the files `thesis.tex` and `appendix1.tex` to see how these pieces work together.

REFERENCES

[1] Anonymous. *The Chicago Manual of Style*. University of Chicago Press, Chicago, IL, USA, 14th edition, 1993.

[2] Anonymous. *Publication Manual of the American Psychological Association*. American Psychological Association, Washington, DC, USA, sixth edition, 2009.

[3] Helmut Kopka and Patrick W. Daly. *Guide to LaTeX: Tools and Techniques for Computer Typesetting*. Addison-Wesley, Reading, MA, USA, fourth edition, 2004. Includes CD-ROM.

[4] Leslie Lamport. *LaTeX: A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, second edition, 1994. Reprinted with corrections in 1996.

BIOGRAPHICAL SKETCH

The author was born, and then the author was "educated," at least to some degree. After finishing high school in Florida, the author completed a Bachelor of Arts degree at The Florida State University. Following a decade in the work force in his discipline, the author returned to FSU to pursue graduate work.