University of Nevada, Reno

This is the title of your graduate thesis or dissertation

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Mathematics

by

You D. Student

Dr. Pat D. Advisor, Ph.D. / Thesis Advisor

May August or December, 20XX

Copyright by You D. Student 20XX

All Rights Reserved.

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

YOU D. STUDENT

entitled

This is the title of your graduate thesis or dissertation

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Pat D. Advisor, Ph.D.
Advisor

Jane Doe, Ph.D.
Committee Member

John Doe, Ph.D.
Committee Member

Sam Doe, Ph.D.
Graduate School Representative

Markus Kemmelmeier, Ph.D., Dean
Graduate School

May August or December, 20XX

Abstract

Enter your abstract here. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend

consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Dedication

I would like to dedicate this work to...

Acknowledgments

I would like to acknowledge...

Table of Contents

Abstract **i**

Dedication **iii**

Acknowledgments **iv**

Table of Contents **v**

List of Tables **vii**

List of Figures **viii**

1 Introduction **1**

1.1 Example: Figure and Table Tagging and Screen Readers 4

1.1.1 Forcing Read Order 5

1.2 `tikz` Example: Morphism objects and operadic centers 7

1.3 Custom Theorem Environment Example 9

References 11

2 The First Project **12**

2.1 Introduction 12

2.2 Results 12

References 16

3 Tips for writing tagging-friendly LaTeX **17**

3.1 Tables 18

3.1.1 Landscape Orientation for Wide Tables & Figures 22

3.2 Equations, revisited 24

3.3 Custom Alternate Text For Math Expressions 26

3.3.1 Screen-Reader Friendly Math Expressions & Symbols 26

3.3.2 Experimental Workaround: Using non-math symbols in math mode 29

Appendices 31

3.A Model Derivation 31

3.B Simulation Details 31

4 Blank Chapter **32**

4.1 Introduction 32

4.2 Results 32

Appendices **33**

A Equations 33

B Simulation Details 34

B.1 Parameter Values 34

B.2 Computer Code 34

List of Tables

3.1 A short table demonstrating a standard LaTeX table. 18

3.2 The longtable package can be used to split a table across multiple pages when it won't fit onto a single page, as illustrated with this table. 20

3.2 The longtable package can be used to split a table across multiple pages when it won't fit onto a single page, as illustrated with this table. 21

3.3 A table of some of the longest words in the Merriam-Webster dictionaries, excluding some molecular and place names. Source: `https://www.merriam-webster.com/wordplay/longest-words-ever`, accessed 10 Apr 2026. . . 23

List of Figures

1.1 The "N" logo for the University of Nevada-Reno. Source: `https://www.unr.edu/Assets/Icons/logos/university-logo.svg` (converted to PDF from SVG) but see also `https://www.unr.edu/marketing-communications/brand/visual-identity`. 2

1.2 The "N" logo for the University of Nevada-Reno. Source: `https://www.unr.edu/Assets/Icons/logos/university-logo.svg` (converted to PDF from SVG) but see also `https://www.unr.edu/marketing-communications/brand/visual-identity`. 6

Chapter 1

Introduction

This example thesis provides some working examples of content that meets current accessibility requirements, assuming you are somewhat familiar with the basics of standard (i.e., pre-2026) LaTeX.

For example, accessibility standards effectively require that documents are able to be read aloud by "screen-reader" software, such as the free software NVDA (a good choice for testing how your documents are read aloud to someone who is visually impaired). Therefore non-text objects like diagrams, figures, tables, and equations like

$$\frac{dx}{dt} = f(x), \quad \text{for} \quad x \in \mathbb{R}^n$$

or navigation links like Figure 1.1 need to be appropriately "tagged" either by automatic structural/MathML tagging by LuaLaTeX or from author-supplied textual descriptions for graphics.

These non-text objects need to be *tagged* in ways that screen-reader software can properly verbalize that content. Fortunately, LaTeX (specifically, LuaLaTeX) will automate the tagging process. For the cases where authors need to assist the tagging process, I'll

Figure 1.1: The "N" logo for the University of Nevada-Reno. Source: `https://www.unr.edu/Assets/Icons/logos/university-logo.svg` (converted to PDF from SVG) but see also `https://www.unr.edu/marketing-communications/brand/visual-identity`.

give some tips below for what you need to do in order to ensure the content that you add to this template is properly tagged and, where necessary, provided with customized (by you) *alternate text* to be read aloud for things that cannot be automated (figure descriptions, for example).

So what do authors need to know? One important thing to know is that there are **ongoing software changes**, and then of course there is the matter of **what content changes need to be made** to how we write the **document header** (e.g., to reflect package usage changes) and **document body content** (e.g., how to add alt text to a figure or ensure a table is properly tagged to indicate that the first row is or is not populated by column headings).

First, regarding the current state of software development, there are multiple variants of TEX and LATEX compiler software, and `pdflatex` has largely been the default compiler that most authors have used over the past few years. The PDFs produced by `pdflatex` typically follow the PDF 1.7 standard, which has limitations for tagging and other internal document structures needed by accessibility software like screen readers. The LuaLATEX compiler, `lualatex`, is similar and uses the Lua programming language to

add some functionality and better interface with other software tools. LuaLaTeX has been around for a few years now, and is a more modern LaTeX engine that provides capabilities not available in pdfLaTeX, including modern Unicode/font support. It was chosen as the development target for upgrading LaTeX to generate PDFs that meet the new PDF 2.0 UA/2 standard, which was designed for better accessibility, and practically works very well as an upgrade for users of pdfLaTeX. **Therefore, to produce an accessible PDF using this template, authors should update to the very latest LaTeX software (typically MiKTeX or TeX Live) and then modify their editor settings or compilation workflow, to use `lualatex` (or the latest development version, `lualatex-dev`, which may require installing the `latex-base-dev` package) instead of `pdflatex`**. This software – and various standard packages – have been very actively developed over the past couple of years, and this activity is ongoing. Therefore, you should make it a habit of updating your LaTeX software very regularly.

Second, regarding content changes, these can be divided into *document header* changes and *document body* changes.

The first big change to the header is the addition of `\DocumentMetadata\{...\}` at the top of the document above `documentclass{...}`, where you can specify settings related to tagging. Next, some packages will no longer be necessary, because certain functionality (e.g., list formatting via the `enumitem` package) is being built into the `lualatex` kernel. Similarly, there are a small number of *new packages* like `unicode-math` that are needed or strongly recommended for generating properly tagging content in your document.

There are a few changes to document body content that may be necessary to learn (e.g., the details of list formatting using the kernel instead of `enumitem` package, which are almost identical!). The changes to how we add content to the body of a LaTeXdocument are fairly minimal (if you already know some LaTeX), which is great news! Experienced

LaTeXauthors will likely have only a few things to learn. Rather than summarize those here, I've included a few examples below, and some additional guidance in the companion document of *LuaLaTeX authoring guidelines* at `https://docs.google.com/document/d/1Bn4H9zB4_EiqsS7m38_1f-udxmJWJgVqUmzK6cKAvJI/edit?usp=sharing`.

1.1 Example: Figure and Table Tagging and Screen Readers

Figures and tables are automatically tagged. To add *alternate text* to your figures is straightforward (see the LaTeX code for Figure 1.1 above): just add an `alt={...}` argument to your `includegraphics` command that loads the figure image. Tables require no alt text, however, they should be limited to having a relatively simple structure (e.g., try to use simple grids, where possible). For more on tables, please see section 3.1 below.

A lesser consideration related to Tables and Figures is the *reading order*. The figure above (Figure 1.1) may not be read by software like NVDA in the reading order expected by sighted readers (you might assume it will be read based on where it appears on the pages). With the current tagging behavior used by this example, deferred floats may occur later in the logical structure than their visual position suggests; in some PDF-reader/screen-reader combinations, they may consequently be encountered much later in the reading order. Thus, it is especially important that tables and figures mentioned in the main text are referenced using the built-in label-reference functionality in LaTeX, with clickable links for accessing them from the main text. This default reading order actually helps visually impaired readers skip over figures and tables until they are mentioned in the main text. If a figure should instead be read in line with the adjacent text, as is commonly the case with something like a commutative diagram, this can be done as described below and the companion document mentioned above.

To summarize what this means for authors:

- We need to add alt text to figures and other graphics, and make sure tables meet accessibility requirements like those spelled out by WCAG standards; and
- Use built-in LaTeX structures and internal referencing, and avoid non-standard formatting "hacks" that aren't tagged properly by these new tools;
- You may, at times, want to turn off tagging for something like a diagram, and specify the alt text yourself, which is discussed below.

Additionally, some informal experimenting with the NVDA screen reader has suggested that it is preferable to reference figures and tables with `\autoref` instead of `\ref`, as the resulting links seem to be better verbalized by screen-reader software.

1.1.1 Forcing Read Order

LuaLaTeX automates tagging very well, however, sometimes you may wish to modify the default tagging behavior for specific objects. For example, suppose you had a small figure that you wanted to force into the linear reading order, and not act as a float.

One way to manually set the location and read order to follow where it appears in the LaTeX source is to put `\tagpdfsetup{float/here}` before the figure, and then after the figure put `\tagpdfsetup{float/defer}` as illustrated by Figure 1.2 (see the LaTeX source for details).

See below for more examples using tables, and other examples that modify the default tagging behavior.

Figure 1.2: The "N" logo for the University of Nevada-Reno. Source: `https://www.unr.edu/Assets/Icons/logos/university-logo.svg` (converted to PDF from SVG) but see also `https://www.unr.edu/marketing-communications/brand/visual-identity`.

1.2 `tikz` Example: Morphism objects and operadic centers

Diagrams can be implemented in two different ways: using code included directly in the document, or loaded from a separate file as with this first example `tikzcd` diagram from Farr (2025). Under earlier versions of LuaLaTeX, these diagrams *needed* to be removed from the main LaTeX source file and compiled on their own (see file `tikzdiagram.tex`) due to packages like `tikz-cd` being incompatible with tagging. The standalone graphic could then be included using the usual image insertion routine from the `graphicx` package: `\includegraphics[alt={alt text}]{figfilename}`. This can still be done. Here is an excerpt from Farr (2025):

> Given a morphism $e : Y \to X$ in $h\mathcal{C}_{\mathfrak{a}}$, the induced map between the fibers comes from solving the lifting problem
>
> $$\begin{tikzcd}
> \{1\} \times \mathrm{Map}_{\mathcal{C}_{\mathfrak{m}}}(X \otimes M, N) \arrow[r, hook] \arrow[d, hook] & \mathcal{C}_{\mathfrak{a}} \times_{\mathcal{C}_{\mathfrak{m}}} \mathcal{C}_{\mathfrak{m}/N} \arrow[d, "f"] \\
> \Delta^1 \times \mathrm{Map}_{\mathcal{C}_{\mathfrak{m}}}(X \otimes M, N) \arrow[ur, dashed] \arrow[dr] & \mathcal{C}_{\mathfrak{a}} \\
> & \Delta^1 \arrow[u, "e"']
> \end{tikzcd},$$
>
> and restricting the lift to $\{0\} \times \mathrm{Map}_{\mathcal{C}_{\mathfrak{m}}}(X \otimes M, N)$. Since f is a pullback of the right fibration $\mathcal{C}_{\mathfrak{m}/N} \to \mathcal{C}_{\mathfrak{m}}$, the lift above is induced by the solution to …

Some diagrams may still give errors, and need to be included as above. For a regular TikZ diagram, you should be able to use a direct inclusion in your document using

```
\begin{tikzpicture}[alt={Diagram description here.}]
 ...
```

```
\end{tikzpicture}
```

Here's a second example diagram, this time made using `tikzpicture` from the `tikz` (not `tikzcd`) package, coded directly in the main document as

```
\begin{center}
  \begin{tikzpicture}[node distance=2cm, auto, alt={Commutative diagram:
   There are four nodes that form the corners of a square. The top left
   is labeled A, the top right B, the bottom left C, and the bottom
   right is labeled D. The arrow from A to B is labeled with a lower
   case f, the arrow from B to D is labeled with h, the arrow from A to
   C is labeled with g, and the arrow from C to D is labeled with k.}]

 % Define the nodes
 \node (A) {$A$};    \node (B) [right of=A] {$B$};
 \node (C) [below of=A] {$C$}; \node (D) [right of=C] {$D$};

 % Draw the arrows and label them
 \draw[->] (A) to node {$f$} (B);
 \draw[->] (A) to node [swap] {$g$} (C);
 \draw[->] (B) to node {$h$} (D);
 \draw[->] (C) to node {$k$} (D);
  \end{tikzpicture}
\end{center}
```

which yields the diagram

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "h"] \\ C \arrow[r, "k"] & D \end{tikzcd}$$

1.3 Custom Theorem Environment Example

Here is an example theorem from Hurtado and Kirosingh (2019). Sometimes a letter-based theorem numbering convention is used in the introduction to introduce a more intuitive statement of a technical theorem that appears later in the document. Here's one such example using a custom `itheorem` environment defined in the preamble of the LaTeX source file for this document. The following theorem is shown in more detail in a subsequent chapter (Theorem 2.1).

Theorem A (Simple LCT (Theorem 2.1)). Consider a continuous time state transition model with inflow rate $\mathcal{I}(t) \geq 0$ into state X which has an Erlang(r, k) distributed dwell time. Let $x(t)$ be the (mean field) amount in state X at time t and assume $x(0) = x_0$. The mean field integral equation for this scenario is

$$x(t) = x_0 S_r^k(t) + \int_0^t \mathcal{I}(s)\, S_r^k(t-s) ds. \tag{1.1}$$

State X can be partitioned into k sub-states X_i, $i = 1, \ldots, k$, where particles in X_i are those awaiting the ith event as the next event under a homogeneous Poisson process with rate r. Let $x_i(t)$ be the amount in X_i at time t, and $x(t) = \sum_{j=1}^k x_j(t)$. Eq. (1.1) is

equivalent to the mean field ODEs

$$\frac{d}{dt}x_1(t) = \mathcal{I}(t) - r\,x_1(t) \tag{1.2a}$$

$$\frac{d}{dt}x_j(t) = r\,x_{j-1}(t) - r\,x_j(t), \quad j = 2, \dots, k \tag{1.2b}$$

with initial conditions $x_1(0) = x_0$, $x_j(0) = 0$ for $j \geq 2$.

References

Farr, Sonja (2025). *$\mathbb{E}_2$-algebra structures on the derived center of an algebraic scheme.* DOI: 10.48550/ARXIV.2506.14069.

Hurtado, Paul J. and Adam S. Kirosingh (Aug. 2019). "Generalizations of the 'Linear Chain Trick': incorporating more flexible dwell time distributions into mean field ODE models." In: *Journal of Mathematical Biology* 79.5, pp. 1831–1883. ISSN: 1432-1416. DOI: 10.1007/s00285-019-01412-w.

Chapter 2

The First Project

Once upon a time...

2.1 Introduction

...

2.2 Results

Here is the more detailed version of Theorem A from Hurtado and Kirosingh (2019).

The Erlang density function (g), CDF (G), and survival function ($S = 1 - G$; also

called the *complementary CDF*) are given by Equation 2.1.[1]

$$g_r^k(t) = r\,\frac{(r\,t)^{k-1}}{(k-1)!}e^{-rt} \tag{2.1a}$$

$$G_r^k(t) = 1 - \sum_{j=1}^{k} \frac{(r\,t)^{j-1}}{(j-1)!}e^{-r\,t} = 1 - \sum_{j=1}^{k} \frac{1}{r}\,g_r^j(t) \tag{2.1b}$$

$$S_r^k(t) = 1 - G_r^k(t) = \sum_{j=1}^{k} \frac{1}{r}\,g_r^j(t). \tag{2.1c}$$

Theorem 2.1 (Simple LCT). *Consider a continuous time state transition model with inflow rate $\mathcal{I}(t) \geq 0$ (an integrable function of t) into state X which has an Erlang(r,k) distributed dwell time (with survival function S_r^k from eq. (2.1c)). Let $x(t)$ be the (mean field) amount in state X at time t and assume $x(0) = x_0$.*

The mean field integral equation for this scenario is

$$x(t) = \; x_0 S_r^k(t) + \int_0^t \mathcal{I}(s)\,S_r^k(t-s)ds. \tag{2.2}$$

State X can be partitioned into k sub-states X_i, $i = 1, \ldots, k$, where particles in X_i are those awaiting the ith event as the next event under a homogeneous Poisson process with rate r. Let $x_i(t)$ be the amount in X_i at time t, and $x(t) = \sum_{j=1}^{k} x_j(t)$. Eq. (2.2) is equivalent to the mean field ODEs

$$\frac{d}{dt}x_1(t) = \mathcal{I}(t) - r\,x_1(t) \tag{2.3a}$$

$$\frac{d}{dt}x_j(t) = r\,x_{j-1}(t) - r\,x_j(t), \quad j = 2, \ldots, k \tag{2.3b}$$

[1] A useful interpretation of survival functions, which is used below, is that they give the expected proportion remaining after a give amount time.

with initial conditions $x_1(0) = x_0$, $x_j(0) = 0$ *for* $j \geq 2$, *and*

$$x_j(t) = x_0 \frac{1}{r} g_r^j(t) + \int_0^t \mathcal{I}(s) \frac{1}{r} g_r^j(t-s) ds. \tag{2.4}$$

Proof. Substituting eq. (2.1c) into eq. (2.2) and then substituting eq. (2.4) yields

$$\begin{aligned} x(t) &= x_0 \, S_r^k(t) + \int_0^t \mathcal{I}(s) \, S_r^k(t-s) \, ds \\ &= x_0 \sum_{j=1}^k \frac{1}{r} g_r^j(t) + \int_0^t \mathcal{I}(s) \sum_{j=1}^k \frac{1}{r} g_r^j(t-s) \, ds \\ &= \sum_{j=1}^k \left(x_0 \frac{1}{r} g_r^j(t) + \int_0^t \mathcal{I}(s) \frac{1}{r} g_r^j(t-s) \, ds \right) = \sum_{j=1}^k x_j(t). \end{aligned} \tag{2.5}$$

Differentiating equations (2.4) (for $j = 1, \dots, k$) yields equations (2.3) as follows.

For $j = 1$, equation (2.4) reduces to

$$x_1(t) = x_0 e^{-r\,t} + \int_0^t \mathcal{I}(s) e^{-r(t-s)} ds. \tag{2.6}$$

Differentiating $x_1(t)$ using the Leibniz integral rule and substituting (2.6) yields

$$\frac{d}{dt} x_1(t) = -r x_0 e^{-r\,t} - r \int_0^t \mathcal{I}(s) e^{-r(t-s)} ds + \mathcal{I}(t) \; = \; \mathcal{I}(t) - r x_1(t). \tag{2.7}$$

Similarly, for $j \geq 2$, one can show

$$
\begin{aligned}
\frac{d}{dt}x_j(t) &= x_0\,\frac{1}{r}\,\frac{d}{dt}g_r^j(t) + \int_0^t \mathcal{I}(s)\frac{d}{dt}\left(\frac{1}{r}\,g_r^j(t-s)\right)\,ds \\
&= x_0\,\left(g_r^{j-1}(t) - g_r^j(t)\right) + \int_0^t \mathcal{I}(s)\left(g_r^{j-1}(t-s) - g_r^j(t-s)\right)\,ds \\
&= r\left(\frac{x_0}{r}\,g_r^{j-1}(t) + \int_0^t \mathcal{I}(s)\frac{1}{r}g_r^{j-1}(t-s)\,ds\right) - \,r\left(\frac{x_0}{r}\,g_r^j(t)\right. \\
&\quad \left. + \int_0^t \mathcal{I}(s)\frac{1}{r}g_r^j(t-s)\,ds\right) = \,r\,x_{j-1}(t) - r\,x_j(t).
\end{aligned}
\tag{2.8}
$$

■

References

Hurtado, Paul J. and Adam S. Kirosingh (Aug. 2019). "Generalizations of the 'Linear Chain Trick': incorporating more flexible dwell time distributions into mean field ODE models." In: *Journal of Mathematical Biology* 79.5, pp. 1831–1883. ISSN: 1432-1416. DOI: 10.1007/s00285-019-01412-w.

Chapter 3

Tips for writing tagging-friendly LaTeX

For more guidance that what is provided below, see the online documents *Creating an Accessible Thesis PDF using LaTeX* at `https://docs.google.com/document/d/1Bn4H9zB4_EiqsS7m38_1f-udxmJWJgVqUmzK6cKAvJI/edit?usp=sharing` and `https://docs.overleaf.com/writing-and-editing/creating-accessible-pdfs`.

There are many ways in which people write certain parts of their LaTeX source files to achieve a desired "look" on the page. However, not all of these tricks provide the kinds of clear structures needed for the new tagging routines to properly tag them. Likewise, people sometimes use certain commands inappropriately, just for formatting purposes, which will give misleading tagging structures (for example, using `\paragraph{}` to format some text is not advised, as this command is basically equivalent to `\subsubsubsection{}`).

Here are some tips and "best practices" for writing tagging-friendly LaTeX:

- **Focus on structure, not just appearance.** Don't use `\\` to force vertical spacing, especially in lists or blockquotes (use alternatives like `\bigskip`, `\smallskip`,

or `\vspace`; I recommend *against* loading the `parskip` package to set spacing as this will very likely alter the thesis template defaults which already conform to the UNR graduate school requirements). To start a new paragraph, use `\par`.

- Use `itemize` or `enumerate`; don't manually type numbered or bulleted lists.
- You are required to add alt text to graphics. Use the `alt` option: `\includegraphics[...,alt={Alt text here}]{filename}`.
- Use ~ for a non-breaking space where a line break would be undesirable; for example, `200~m` keeps the value and unit together.
- Use `\paragraph{}` appropriately, as mentioned above.
- **Mathematics/Equations tip:** Do not use the older TeX syntax `$$...$$` for *display math* mode (i.e., an equation centered on its own line). You should instead use `\[ ... \]` for display math. You could use `\( ... \)` for inline math, but it's also OK to use `$...$`. Similarly, please use proper equation environments like `align` when those are more appropriate than shortcuts like using multiple 1-line equations using consecutive `\[ ... \]` environments.

3.1 Tables

Table 3.1: A short table demonstrating a standard LaTeX table.

Letter	**Number**
A	1
B	two
C	3

Here's a simple table, Table 3.1, however, based on my testing with NVDA in mid-2026, it likely won't be read aloud by a screen reader until the very end of the doc-

ument unless the reader follows a link in the main text, or unless the author manually forces a change to the read order as illustrated above with Figure 1.2. Float objects, like tables and figures, are not read in the order they appear on the page by default. Additionally, links to a standard table (or figure) may not be verbalized well, again, because of the way floats are tagged. Experiment with using `\autoref{...}` vs. `\eqref{...}` vs. `\ref{...}`; in my testing `\autoref{...}` might be read better by software like NVDA compared to `\ref{...}`. The default behavior of tables, figures, and other floats, should be used unless there is a strong reason to change the read order.

Next, here's a table that is only used for text formatting and is therefore manually configured to be tagged differently from standard tables like the one above (see the LaTeX source for details). It should be read by a screen reader as plain text.

Name:	John Doe
Date:	May 2025

The `longtable` package allows tables to be split across multiple pages, and has recently gained what I would call partial compatibility with tagging (but do be careful if you use the `caption` package, as it may still cause problems). Note that, unlike a standard table, one must use a different approach to tell the tagging procedure that column labels are in the first row of the table by adding `\endhead` to the end of the table header row. For example,

```
\LTcapwidth=\textwidth %% Do NOT use the next line...
%\tagpdfsetup{table/header-rows={1}}
% The line above is IGNORED if \endhead or \endfirsthead are used.
\begin{longtable}{|c|c|}
 \caption{...}\label{tblname} \\
 \hline
 \textbf{Base 10} & \textbf{Binary} \\
```

```
 \hline \endhead  %% Tag the row(s) above as the header row.
 ...
\end{longtable}
```

Please read through the longtable package documentation carefully if you use a `longtable`, especially the section on *Captions and Headings* and any sections that mention tags and tagging.

Note: The `caption` is treated as the first row, spanning all columns, via the use of `\multicolumn`. In my testing, I've found that the Adobe checker flags this as a *Regularity* error. This **seems to be an overly strict interpretation of the PDF UA/2 and WCAG standards**: The NVDA screen reader seems to correctly read tables like the one below. This *Regularity* failure might therefore be safe to ignore, especially if the veraPDF checker gives it a pass (further testing needed). This overly strict interpretation of the accessibility standards by the Adobe checker is a known issue, among others, as noted at `https://github.com/latex3/tagging-project/discussions?discussions_q=is%3Aopen+label%3A%22spurious+validation+error+%5BAdobe%5D%22`.

Table 3.2: The longtable package can be used to split a table across multiple pages when it won't fit onto a single page, as illustrated with this table.

Base 10	**Binary**
1	1
2	10
3	11
4	100
5	101
6	110

Table 3.2: The longtable package can be used to split a table across multiple pages when it won't fit onto a single page, as illustrated with this table.

Base 10	**Binary**
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000
17	10001
18	10010
19	10011
20	10100
21	10101
22	10110
23	10111
24	11000
25	11001

3.1.1 Landscape Orientation for Wide Tables & Figures

Sometimes, wide tables are best shown rotated 90°, so that the table appears on the page in landscape (rather than portrait) orientation. The page itself can then be displayed in a PDF reader in the normal orientation (portrait) or rotated by the PDF viewer to landscape for better viewing of the table.

Packages that do this are `lscape` (which does not rotate the page in the PDF, just the table on the page) and `pdflscape`, which will also rotate the page in the PDF. You'll typically want to use `pdflscape`, unless you specifically don't want the PDF page rotated for viewing on screen.

This next page shows Table 3.3, an example using `pdflscape`, and the `tabularx` package to fill out the full space available on the page.

Table 3.3: A table of some of the longest words in the Merriam-Webster dictionaries, excluding some molecular and place names. Source: `https://www.merriam-webster.com/wordplay/longest-words-ever`, accessed 10 Apr 2026.

Number of Letters	**Word**	**Description/Definition**
45	pneumonoultramicroscopic-silicovolcanoconiosis	"A lung disease caused by inhalation of very fine silicate or quartz dust."
30	pseudopseudohypoparathyroidism	"Refers to a relatively mild form of pseudohypoparathyroidism." "The pathophysiology of PFBC is profoundly linked to the loss of integrity of the BBB and should be distinguished from the calcification processes that take place in secondary brain calcifications, in which calcium deposition develops via two main pathways: dystrophic calcification resulting from membrane disruption and uncontrolled calcium entry (hypoxic ischemic injury, intrauterine infections), and calcium–phosphate metabolism alterations (pseudohypoparathyroidism and pseudopseudohypoparathyroidism, mainly linked to loss of function mutations of GNAS, involved in the intracellular transmittal pathway of PTH). — Monfrini et al., *International Journal of Molecular Sciences (Basel, Sz.)*, 2023"
27	electroencephalographically	Indicates something done "using an apparatus for recording brain waves, that is, an electroencephalograph."
26	radioimmunoelectrophoresis	"Refers to a kind of immunoelectrophoresis involving radioactive labels."
25	immunoelectrophoretically	To do something with immunoelectrophoresis.
24	laryngotracheobronchitis	"Refers to inflammation of the larynx, trachea, and bronchi."

3.2 Equations, revisited

It's worth repeating: to create display mode math expressions, please do not use the older TeX double-$ notation like `$$f(x)=x^2$$`. Instead use `\[f(x)=x^2\]`. For inline math, you *can* use either `\(...\)` (recommended), or the more common `$...$` (also acceptable).

Here are a few example equations using different environments. Know how to use different environments to get multiline equations that are either labeled as separate equations, labeled as subequations, or labeled as a single equation spanning multiple lines. See the AMS Short Math Guide for LaTeX at `https://tug.ctan.org/info/short-math-guide/short-math-guide.pdf` for more examples than those given below.

Using the `align` environment to position the equations relative to one another, we have different options for how they are labeled and assigned equation numbers. For example, used alone, an align environment gives eqs. (3.1).[1]

$$\dot{x} = \frac{dx}{dt} = f_x(x, y) \tag{3.1}$$

$$\dot{y} = \frac{dy}{dt} = f_y(x, y). \tag{3.2}$$

To get the above, but numbered as subequations, use `align` within a `subequations` environment, where you can label the set of equations as well as individual equations:

$$\dot{x} = \frac{dx}{dt} = f_x(x, y) \tag{3.3a}$$

$$\dot{y} = \frac{dy}{dt} = f_y(x, y). \tag{3.3b}$$

This way, you can refer to the system of equations (3.3), or the individual Equation 3.3a and Equation 3.3b.

[1]The abbreviation "eqs." is also probably not very screen reader friendly, so it is preferred to use `\autoref*{label}` instead, which will give "Equation 3.1" or maybe find a way to make a macro that looks like `\Eqs` and reads as "equations".

Here is an aligned environment in an equation environment:

$$\begin{aligned} \frac{dx}{dt} &= D_t x \\ &= f_x(x, y). \end{aligned} \tag{3.4}$$

Some bold math (note how these are read by a screen reader like NVDA, then see the next section).

$$\mathbf{A} = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix}_{2\times 2}$$

Equation referencing note: To reference equations in the main text or figure and table captions, you have a few options. First, using `\autoref{label}` from the `hyperref` package will work to give you something like "Equation 1" where the text and number are a clickable link (use `\autoref*{label}` to suppress links). Second, many journals require the number to be in parentheses, and/or they require the use of abbreviations Eq. or Eqs., so that is more standard notation. Therefore, you might instead choose to use the more traditional "`Eqs.~\eqref{label}`." Here the text label "Eqs." won't be part of the clickable link: for example, "Eqs. (3.3)". Optionally (my recommendation), define a custom referencing command like `\newcommand{\Eqs}[1]{\hyperref[#1]{Eqs.~(\ref*{#1})}}` in your preamble. This makes "Eqs." part of the clickable link, and helps ensure that you consistently use ~ for the space, and thus helps `lualatex` tag equation references so they are verbalized well by screen readers like NVDA. Using `\Eqs{label}` gives linked labels like Eqs. (3.3).

3.3 Custom Alternate Text For Math Expressions

You may encounter situations where you'd like to modify the standard alternate text for math expressions. For example, there are some mathematical symbols that either do not have Unicode character analogs or that we would prefer to manually assign some context-specific alternate text that differs from the standard MathML semantics. For context, the `unicode-math` package helps the new MathML generation/tagging machinery to automatically tag math symbols like α with the corresponding Unicode character, rather than the LaTeX "`\alpha`", which ensures that such symbols can be read properly by screen-reader software like NVDA.

In the January 2026 release of LuaLaTeX, the new MathML `intent` attribute was integrated into the MathML-SE tagging setup to allow authors to specify custom text for symbols and expressions. For example, to use context-specific semantics like a more descriptive/interpretive name for a symbol. Note that the syntax used here may change as the LaTeX Tagging Project matures, or may be surpassed by alternative approaches.

These next two sections look at customizing the alternate text related to how the MathML tags are generated using the MathML `intent` attribute.

3.3.1 Screen-Reader Friendly Math Expressions & Symbols

The LaTeX tagging routines that convert equations to MathML don't fully grasp all of the contextual nuances that we use to read math. This leads screen readers to read some expressions incorrectly. For example,

$$f'(f'(x))$$

will get read by NVDA as something like "f prime times f prime of x" and this is a com-

mon problem with functional expressions where the function or the argument isn't just a single character! One solution is to make a new command using `\MathMLintent{}{}` that is a proper function and has custom alternate text, so that it is read aloud correctly.

Example 1

To see how to implement such improvements, see the LaTeX source for this section, and the custom commands/macros in the preamble of the document (I've also included these below). Also compare how a screen reader like NVDA reads the following two expressions. These are visually identical, but are read differently by the NVDA screen reader. The first is based on standard LaTeX math notation and suffers from the same problem as the above example as well as the left hand side being poorly verbalized.

$$(f^{-1})'(y) = \frac{1}{f'(f^{-1}(y))}.$$

The following alternative is read aloud with user-defined alternate text:

$$\MathMLintent{f-inverse-prime($x)}{{(f^{-1})'\!\left(\MathMLarg{x}{{#1}}\right)}}.$$

The second example uses these functions, defined in the document preamble:

```
\newcommand{\finvprime}[1]{%
 \MathMLintent{f-inverse-prime($x)}%
 {{(f^{-1})'\!\left(\MathMLarg{x}{{#1}}\right)}}%
}% \finvprime{y} is read by a screen reader as "f inverse prime of y"

\newcommand{\finv}[1]{%
 \MathMLintent{f-inverse($x)}%
 {{f^{-1}\!\left(\MathMLarg{x}{{#1}}\right)}}%
}% \finv{y} will be read by a screen reader as "f inverse of y"

\newcommand{\fprime}[1]{%
 \MathMLintent{f-prime($x)}%
```

```
  {{f'\!\left(\MathMLarg{x}{{#1}}\right)}}%
 }% \fprime{\sin x} is read as "f prime of sin of x"
```

Importantly, these functions can be given more complex mathematical expressions as arguments, and will still be verbalized correctly by a screen reader like NVDA.

Remark 1: The second argument to `\MathMLintent{}{}` needed to include an extra pair of curly braces to compile correctly when the argument includes more than one math symbol!! This may change as the software develops.

Remark 2: The mathml-SE tagging method is preferred over AF tagging, as it provides more consistency across screen-reader software and PDF readers. Some PDF and screen reader combinations, especially when configured with different settings, may handle the combined SE and AF tagging methods differently. There is a trade-off, however: Using both mathml-SE and mathml-AF will help ensure the document can be read by older screen-reader software, at the cost of possible inconsistencies.

Example 2

Authors often encode information visually using different fonts (e.g., $\mathcal{F}$ versus F), and `\newcommand{}{}` and `\MathMLintent{}{}` can similarly be used to give math symbols and functions custom alternate text. For example, to define a boldface letter J to be read as a Jacobian ($\mathbf{J}$), and a regular J to be read as the Poisson matrix (J), one can use the following:

```
% Example: give standard symbols context-specific alt text
\newcommand{\Jac}{%
 \MathMLintent{Jacobian-J}%
 {\mathbf{J}}%
}
\newcommand{\J}{%
 \MathMLintent{Poisson-matrix-J}%
 {J}%
}
```

Example 3

Sometimes, we just want to use context-specific semantics instead of default MathML. Consider the double-struck "1" symbol, $\mathbb{1}$, which is often used as an indicator function that has value 0 or 1 depending on whether its argument belongs to a given set. Unfortunately, the default MathML reading of the symbol leads screen readers to read it as "double-struck one."

In the LaTeX source file for this document I have added two commands/macros, above the `\documentclass{}` line, that define a new symbol and a new math function that can be used as one would use, say, `\pi` and `\sqrt{...}`:

```
\newcommand{\indAfn}{% Verbalized as "indicator function subscript A"
 \MathMLintent{indicator-function-subscript-A}%
 {{\mathbb{1}_A}}%
}
% Below is verbalized as "indicator function subscript A of [arg]"
\newcommand{\indAfnof}[1]{
 \MathMLintent{indicator-function-subscript-A($x)}%
 {{\mathbb{1}_A\!\left(\MathMLarg{x}{{#1}}\right)}}%
}
```

Using these commands, NVDA read aloud the phrase "$\mathbb{1}_A$ is 0 or 1, where $\mathbb{1}_A(\omega)$ is 1 if $\omega \in A$ and 0 otherwise" as "indicator function subscript A is zero or one, where indicator function subscript A of omega is one if omega is an element of A and zero otherwise."

3.3.2 Experimental Workaround: Using non-math symbols in math mode

Sometimes we want to broaden our mathematical symbols to include what the LaTeX tagging routines would treat as non-mathematical objects. Giving these non-math symbols alternate text within the context of MathML tagging is more difficult. Here is a

workaround if you wanted to try and implement custom symbols with custom alternate text.

For example, 🛠 (a hammer and wrench crossed in an X pattern) is given by `\usym{1F6E0}` in the `utfsym` package, and has no associated alt text or MathML recognition. We can use this as a new math symbol and/or an operator that takes an argument, as in the above examples, by specifying a pair of macros: one for the symbol and one for the function form. As in the previous example, each includes alternate MathML text. Unlike the previous example, the symbol macro does not work without first fooling the LaTeX MathML tagging routine into recognizing it as a math object: Here, this was done by placing the custom symbol over a `\cdot` symbol. Presumably, properly declaring it as a new math symbol somehow would also work (and be the preferred approach).

```
\usepackage{utfsym}
\newcommand{\hammerwrench}{% Symbol only
 \MathMLintent{hammer-wrench}%
 %{{\usym{1F6E0}}}% Doesn't work! Argument needs to be a math object
 {{\,\cdot\!\!\!\text{\usym{1F6E0}}}}% This line works!! Why?
 % Placing it over a \cdot coerces the 2nd argument to a math object!
 % Declaring it a new math object somehow should also work.
}
\newcommand{\hammerwrenchof}[2]{% Function version that takes 2 arguments
 \MathMLintent{hammer-wrench($x,$y)}%
 {{\usym{1F6E0}\!\left(\MathMLarg{x}{{#1}},\,\MathMLarg{y}{{#2}}\right)}}%
}% Maybe try \mathop{} if the symbol isn't behaving well in math mode
```

Now we can see that the 🛠 function, perhaps written as 🛠$(a,\, b)$, can be used in expressions like $\sin\left(\sqrt{\text{🛠}(\alpha,\, \omega)}\right)$ and will be read properly as "sine of, open paren, the square root of, hammer-wrench of alpha and omega, close paren" by a screen reader, e.g., using Adobe with NVDA.

Appendices

3.A Model Derivation

These are just some example appendices at the end of the chapter instead of at the end of the thesis.

We just nondimensionalized the equation $\dot{N} = r\,N\,(1 - N/K)$ to get $\dot{x} = x(1 - x)$.

3.B Simulation Details

We used Euler's method with step sizes of 10^{-4} and initial condition $x(0) = 0.1$.

Chapter 4

Blank Chapter

Some more content...

4.1 Introduction

Stuff

4.2 Results

QED.

Appendices

Example appendices at the end of the thesis instead of individual chapters.

A Equations

Here is the identity matrix,

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{A.1}$$

B Simulation Details

B.1 Parameter Values

We used the values in the code below.

B.2 Computer Code

Here's some R code. Unfortunately, packages like `listings` are not yet compatible as of early October 2026, with the new LaTeX tagging functionality, so you may have to wait a while before you can submit nicely formatted code with syntax highlighting. Until then, we can use a trusty old `verbatim` environment. Better yet: code should be an electronic supplement to your thesis. Code opened in a screen reader friendly code editor will be much more accessible than code included in your thesis PDF.

R code: Example Script

```
# THIS IS A LOVELY LITTLE BIT OF R CODE:
# --------------------------------------
# install.packages("openssl") # install this first, then run the code below
par(bg = "black", fg = "black")
x=seq(-sqrt(3),sqrt(3),length=800)
for(k in seq(0,100,length=40)) {
 plot(x,(x^2)^(1/3)+0.9*sin(k*x)*sqrt(3-x^2), type="l",lty=1, col="red",
 xlim=c(-2,2), ylim=c(-1.25,2.25), lwd=2)
 text(0,1,
   rawToChar(openssl::base64_decode("SGFwcHkgVmFsZW50aW5lJ3MgRGF5IQ==")),
   col="white", cex=3.25)
 Sys.sleep(1)
} # End of Example Script
```

Remark: If you want computer code that is better formatted, you might be able to achieve this by modifying your LaTeX editor settings so that it can use R to compile `knitr` *.Rnw files (LaTeX documents with blocks of R code, *or code from other languages*) which can yield nicely formatted code without using the (currently incompatible) `listings` or `minted` packages. Rather than try to add a detailed introductory knitr/Sweave tutorial to this example thesis document, see the examples and software guidance to get you started at `https://pauljhurtado.com/latex/` and also check `https://docs.google.com/document/d/1Bn4H9zB4_EiqsS7m38_1f-udxmJWJgVqUmzK6cKAvJI/edit?usp=sharing`.