

THESIS TITLE

John Doe

A THESIS

Presented to the Faculty of Miami University in partial
fulfillment of the requirements
for the degree of

Master of Science

Department of Electrical and Computer Engineering

The Graduate School
Miami University
Oxford, Ohio

2024

Advisor 1, Advisor
Advisor 2, Advisor
Reader 1, Reader
Reader 2, Reader

©

John Doe

2024

ABSTRACT

This document is a template and guide for writing a thesis in \LaTeX . The document is divided into sections explaining the preferred way of typesetting equations, figures, tables, and more. To prepare your thesis, simply edit the file `thesis.tex` along with the chapters in the `chapter/` folder. To make the final pdf, open a command prompt, navigate to the folder containing the template, and run the following commands:

```
pdflatex thesis.tex
bibtex thesis.aux
pdflatex thesis.tex
pdflatex thesis.tex
```

Table of Contents

List of Tables	v
List of Figures	vi
Dedication	viii
Acknowledgements	ix
Acronyms	x
1 Introduction to L^AT_EX	1
1.1 Section title	1
1.2 Compiling	2
2 Mathematics	4
2.1 Typesetting equations	4
2.2 Linear algebra	5
3 Environments	7
3.1 Theorems	7
3.2 Lists	7
4 More	9
4.1 References and links	9
4.2 Figures	9
4.3 Tables	9
4.4 Code	10

A Background	12
References	13

List of Tables

4.1	Gravimetric analysis of silver halides in a 1.27-mL sample of sea water. . . .	10
-----	--	----

List of Figures

4.1	Figure captions should be long and descriptive because people actually read them, unlike the rest of the text. (Left) A block diagram make using Tikz. (Right) A plot made using Pgfplots. Note that the figures are not scaled, so the text size in the figures is consistent with the rest of the document. The source files for the figures are in the <code>graphics/</code> folder.	10
-----	--	----

Dedication

I would like to dedicate this thesis to...

Acknowledgements

I would like to acknowledge...

Acronyms

KYP Kalman–Yakubovitch–Popov

TCP/IP Transmission Control Protocol/Internet Protocol

1

Introduction to L^AT_EX

LaTeX is a typesetting program used to create high-quality documents¹. It is free to use, and is based on the typesetting system TeX created by Donald Knuth in 1978. Nearly all professional papers are created using LaTeX, and any research papers submitted to IEEE should be written using it. While there is a bit of a learning curve to get going, you will find that it is a very elegant and powerful program once you have mastered it!

Standard word processors — such as Microsoft Word — are called *what you see is what you get*, meaning that you directly edit the final document with all of its final formatting. In contrast, LaTeX separates the content from its layout. You write content in `.tex` files, which are then processed by LaTeX to produce the final document, typically a pdf. This frees you from the burden of thinking about the final layout of the document while you write. For example, this template is written using the custom `thesis.cls` style file which tells LaTeX how to format the document to meet the requirements of a thesis. But you do not need to know anything about the formatting to use the template. You just fill in your content and let LaTeX take care of building the final document for you. You are encouraged to look at the source code of this document for reference, since it was made using LaTeX.

To learn more about LaTeX, see some of the many references online².

1.1 Section title

You can make sections using the `\section` command. Sections are automatically added to the table of contents (but you need to run `pdflatex` twice for them to update!).

Put a blank line between text to put it in a separate paragraph.

¹<https://www.latex-project.org/about/>

²https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes

Paragraph title. The `\paragraph` command can be used to break up long blocks of text and help the reader. Paragraph titles should have the first word capitalized and then the others lowercase, and they should end with a period.

Acronyms. We should try to avoid acronyms wherever possible. The only acronyms that should be used are those that are so common that they are more easily recognized than their definitions, such as “the Kalman–Yakubovitch–Popov (KYP) lemma” or Transmission Control Protocol/Internet Protocol (TCP/IP). As Richard Murray is fond of saying, defining an acronym just to save typing in a paper is very bad for a reader. We can read a familiar sequence of words almost as fast as we can read an acronym, and much faster if we have to pause to remember what the acronym is actually for. Generally speaking, we should limit ourselves to at most one or two new acronyms, and they should be reserved for things like algorithm names that we hope other people will also use. To organize acronyms, you can use the `acronym` package. For example, you can use the acronyms defined above as KYP and TCP/IP.

1.2 Compiling

To build the final document, you need to compile the `.tex` files. You can do this by installing LaTeX on your computer using a distribution such as MikTeX³, or using the online LaTeX editor Overleaf⁴. If you choose to download LaTeX, a good editor is Visual Studio Code⁵, which has a LaTeX extension⁶ that provides code highlighting and easy compilation. You can also build the files from the command line using the following:

```
pdflatex thesis.tex
bibtex thesis.aux
pdflatex thesis.tex
pdflatex thesis.tex
```

It may seem odd to have to rerun the command so many times, but each pass stores certain auxiliary information (if you look in the main folder after running these commands, there are many new files such as `.aux`, `.lof`, `.log`, `.lot`, `.out`, and `.toc`). These files contain the information required to make things like the table of contents, list of figures, list of tables, references, etc. The first command creates the auxiliary file `thesis.aux` that indicates

³<https://miktex.org/>

⁴<https://www.overleaf.com/>

⁵<https://code.visualstudio.com/>

⁶<https://marketplace.visualstudio.com/items?itemName=James-Yu.latex-workshop>

which references are cited. The second command runs `bibtex` on the file `thesis.aux` to create the bibliography in `thesis.bbl`. The third command then builds the document with the bibliography, and the bibliography labels are written into the `.aux` file. And finally, the compiler has access to all the required information on the last run to build the final document correctly.

In a typical workflow, you do not need to rerun all of these commands every time you edit the document. Typically, you just need to rerun `pdflatex` to update the pdf. If there were errors, however, you will need to rerun `bibtex` to fix the bibliography. Make sure to run all four commands to build the final document that you submit!

These four commands are in the file `make.bat`, so you can also run that file to make the final document.

1.2.1 This is a subsection

You can organize your document using subsections.

This is a subsubsection

And subsubsections...

This is a paragraph. And even paragraphs. Note that paragraphs continue on the same line as the heading, and there is a period automatically inserted after the heading. By default, subsections, subsubsections, and paragraphs are not included in the table of contents.

2

Mathematics

One of the main benefits of LaTeX is that it produces beautiful equations. You may think that you can replicate this in Microsoft Word, but I assure you, any mathematician can tell the difference (and loathes nasty pixelated equations!).

2.1 Typesetting equations

For equations, we take our guidelines from the `amsmath` package. Inline equations such as $e^{i\pi} = -1$ should be typeset using the `$...$` syntax.

Displayed equations. If an equation is larger or more important and you want to put it on its own line, use the `equation` or `equation*` environment, depending on whether you want equation numbers or not. For non-numbered equations, you can use the shortcut `\[...\]`. Only number an equation if you plan on referring to it later on. Equations are part of the text, so use punctuation in equations. To refer to an equation, give it a `label` and use the `eqref` command to refer to it. For example, the inline equation above was

$$e^{i\pi} + 1 = 0, \tag{2.1}$$

and we can refer to it as (2.1) in the main text. Do not put the word “equation” before the reference, unless it starts a sentence. Equation (2.1) is an elegant equation that uses the symbols e , i , π , 0 , 1 , and $+$ all in a single equation. When starting a sentence with an equation, you can either type the word “equation” or use the `\Cref` command. See Section 4.1 for more discussion about references.

Long equations. For equations that do not fit on one line, use the `multline` or `multline*` command, again depending on whether you want an equation number or not. For example,

$$\begin{aligned} (x + y + z)^4 = x^4 + 4x^3y + 4x^3z + 6x^2y^2 + 12x^2yz + 6x^2z^2 + 4xy^3 + 12xy^2z \\ + 12xyz^2 + 4xz^3 + y^4 + 4y^3z + 6y^2z^2 + 4yz^3 + z^4. \end{aligned} \quad (2.2)$$

When breaking apart an equation, put the dangling symbol (in the case above, $+$) on a new line. Do not end a line with a symbol.

Aligned equations. To align equations, use the `align` environment. A common usage is to align equations at the $=$ sign to make them more legible. When numbering equations that belong to the same group, use the `subequations` environment. This produces

$$\dot{x}(t) = Ax(t) + B_1u(t) + B_2w(t), \quad (2.3a)$$

$$y(t) = C_1x(t) + D_{11}u(t) + D_{12}w(t), \quad (2.3b)$$

$$z(t) = C_2x(t) + D_{21}u(t) + D_{22}w(t). \quad (2.3c)$$

We can then refer to all equations together as (2.3), or to individual equations such as (2.3a) and (2.3b).

Additional information. For a detailed guide on how to typeset equations, it is worth taking a look at the `amsmath` user's guide¹.

2.2 Linear algebra

Vector spaces. The vector space of n -dimensional real vectors is \mathbb{R}^n . Use the commands `\bmat` and `\pmat` to make vectors (and matrices) that use square brackets and parentheses, respectively. For example,

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad \begin{pmatrix} 4 \\ 3 \\ -1 \\ 2 \end{pmatrix}.$$

The set of complex numbers is \mathbb{C} , and the set of integers is \mathbb{Z} .

¹<http://mirrors.ctan.org/macros/latex/required/amsmath/amslatex.pdf>

Matrices. Matrices are denoted by capital letters, such as A or B . The transpose of a matrix is A^\top , and the conjugate transpose is A^* . For square matrices, the trace is $\text{tr}(A)$ and the determinant is $\det(A)$. Like vectors, you can create matrices such as

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}.$$

To separate blocks of a large matrix, use the `\array` environment, and use `\vdots`, `\ldots`, and `\ddots` to indicate repeated patterns in matrices. For example,

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] = \left[\begin{array}{cccccc|c} 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ \hline -a_0 & -a_1 & -a_2 & \dots & -a_{n-2} & -a_{n-1} & 1 \\ b_0 & b_1 & b_2 & \dots & b_{n-2} & b_{n-1} & 0 \end{array} \right].$$

3

Environments

Environments in LaTeX are used to apply specific formatting to part of the document¹.

3.1 Theorems

For Theorem environments, we use the `amsthm` package. This allows us to define environments that are frequently used such as `thm` for theorem, `lem` for lemma, and so on. We can also use the `thmtools` package to create boxed or shaded theorems for emphasis. Here are some examples.

Theorem 1 (A theorem). This is how we state a theorem.

Theorem 2 (Shaded). For emphasis, we can put it in a shaded box.

Theorem 3 (Outlined). Another way to create emphasis is with an outlined box.

Proof. We can write proofs using the `proof` environment. ■

The `thmtools` package also provides `restatable`, which is useful if you want to state the same result more than once (say, in the introduction and later in the paper), but don't want to give it a new label and equation numbers. See the documentation for more details².

3.2 Lists

Create bulleted lists using the `itemize` environment. For example:

- First item

¹<https://www.overleaf.com/learn/latex/Environments>

²<https://ctan.math.illinois.edu/macros/latex/contrib/thmtools/doc/thmtools-manual.pdf>

- Second item
- Third item

Numbered lists are created using the `enumerate` environment. For customization, we use the `enumitem` package with the `shortlabels` option. This allows us to write customized lists easily. For example,

<code>\begin{enumerate}[(i)]</code>		
<code>\item first item \label{x}</code>		(i) first item
<code>\item second item \label{y}</code>	produces	(ii) second item
<code>\item third item \label{z}</code>		(iii) third item
<code>\end{enumerate}</code>		

We can refer to items using `\cref` as before. For example, the command `\cref{x,y,z}` produces Items (i) to (iii).

4

More

4.1 References and links

We use the `hyperref` package to produce a pdf with hyperlinks. We also use the `cleveref` package for facilitating references. Using the `\cref` command will automatically use the correct prefix. You can also refer to multiple things at once by using multiple arguments, or you can refer to a range using `\crefrange`. For more information, see the documentation¹.

References can be cited with the `\cite` command, which produces something like [1–4]. The `cite` package ensures the citations are ordered nicely and compressed when possible.

4.2 Figures

We use the standard `figure` environment for figures. Diagrams should be placed in separate files in the `graphics/` folder and should use the `standalone` package. You can then use `includegraphics` as in Figure 4.1 to include the figure in your document. If you are using `pdflatex`, you can also use `includegraphics` to include PDF, JPEG, PNG, or other formats, but not EPS.

4.3 Tables

The `booktabs` package, which includes commands such as `\toprule`, `\midrule`, and `\bottomrule`, can be used to make nice tables. In general, never use vertical lines to separate columns. For more style tips on how to make nice tables, see². Here is an example of a nice table³.

¹<http://mirrors.ctan.org/macros/latex/contrib/cleveref/cleveref.pdf>

²<https://people.inf.ethz.ch/markusp/teaching/guides/guide-tables.pdf>

³<https://lazyscientist.wordpress.com/2021/07/23/make-better-tables-in-latex-using-booktabs/>

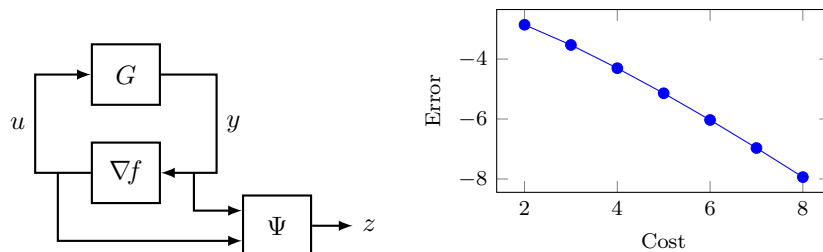


Figure 4.1: Figure captions should be long and descriptive because people actually read them, unlike the rest of the text. (Left) A block diagram made using Tikz. (Right) A plot made using Pgfplots. Note that the figures are not scaled, so the text size in the figures is consistent with the rest of the document. The source files for the figures are in the `graphics/` folder.

Table 4.1: Gravimetric analysis of silver halides in a 1.27-mL sample of sea water.

Qty of Sample	Test Tubes				Avg
	A	B	C	D	
Mass (g)	1.399	1.32	1.328	1.408	1.364
Density (g/mL)	1.10	1.04	1.05	1.109	1.07
Mass w/ Precipitate (g)	13.443	13.401	13.348	—	13.397
Mass AgCl (10^{-2} g)	9.0	9.2	8.7	—	8.9
Moles AgCl (10^{-4} mol)	6.28	6.42	6.08	—	6.50

4.4 Code

We can include snippets of code when appropriate to describe specific computer code. A sample program in Julia is shown in Listing 1. Note that this uses the `pygmentize` package in Python, which must both be installed in order to compile. Also, the code contains unicode characters, which must be defined (see the preamble of `thesis.tex`).

```

1  m,L = 1,10                                # parameters
2   $\alpha$  = 2/(L+m)                          # stepsize
3
4  @algorithm begin
5      f = DifferentiableFunctional{R^n}()      # objective function
6      xs = first_order_stationary_point(f)      # first-order stationary point
7      f'  $\in$  SectorBounded(m, L, xs, f'(xs))    # function class
8      x0 = R^n()                              # initial point
9      x1 = x0 -  $\alpha$ *f'(x0)                   # gradient descent
10     x0 => x1                                  # state update
11     performance = (x1-xs)^2                  # performance measure
12 end
13
14 rate(performance)                          # automated algorithm analysis

```

Listing 1: A sample code listing. The method `rate` performs the automated algorithm analysis by searching for a Lyapunov function that bounds the rate of convergence of the performance measure.

A

Background

Use appendices to provide background information that the reader may already know, but is useful in case they need a refresher. A thesis on control, for example, may have an appendix on the linear–quadratic regulator, or some linear algebra.

References

- [1] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.
- [2] Laurent Lessard and Sanjay Lall. Optimal control of two-player systems with output feedback. *IEEE Transactions on Automatic Control*, 60(8):2129–2144, 2015.
- [3] Akhil Sundararajan, Bryan Van Scoy, and Laurent Lessard. Analysis and design of first-order distributed optimization algorithms over time-varying graphs. *IEEE Transactions on Control of Network Systems*, 7(4):1597–1608, 2020.
- [4] Bryan Van Scoy, Randy A Freeman, and Kevin M Lynch. The fastest known globally convergent first-order method for minimizing strongly convex functions. *IEEE Control Systems Letters*, 2(1):49–54, 2017.