THESIS TITLE

by

First Middle Last

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Chemical and Materials Engineering
University of Alberta

Abstract

This guide is a comprehensive resource crafted for University of Alberta students tackling their theses with LaTeX. It is designed to simplify the thesis-writing process by offering a detailed walkthrough of a custom LaTeX template tailored to meet the university's formatting requirements. The guide goes beyond mere template usage, providing step-by-step instructions, tips, and best practices for creating a well-structured thesis that adheres to academic standards.

It also delves into advanced customization techniques, allowing users to achieve specific stylistic elements and formatting nuances that suit their individual needs. Alongside these, the guide introduces JabRef, a robust reference management tool, and explains how to seamlessly integrate it with LaTeX to streamline citation management. Whether you are a novice or an experienced LaTeX user, this guide equips you with the knowledge and tools necessary to produce a polished, professional thesis. By following its clear and concise instructions, students can confidently navigate the complexities of thesis writing, ensuring their work is both technically sound and perfectly laid out. Test.

Preface

Writing a thesis is no small task, and as a graduate student at the University of Alberta, I quickly realized just how challenging it can be to meet all the formatting requirements while also producing a document that looks professional. Like many others, I started out using traditional word processors, but it didn't take long before I ran into the usual headaches—crashes, file corruption, and formatting issues that seemed to have a mind of their own.

These frustrations led me to explore LaTeX as an alternative. I discovered that LaTeX not only provided a way to keep my content and formatting separate but also offered a more reliable and consistent way to produce a high-quality thesis. The learning curve was steep, but once I got the hang of it, I found it to be a game-changer.

The original template document was born out of my experience with LaTeX and the desire to make the thesis-writing process a bit less daunting for others who might be in the same boat. My goal here with this new document is to provide a comprehensive guide that not only walks you through the basics of LaTeX but also gives you practical examples and best practices[1] to follow.

I built this template and class from the ground up with the aim of reducing the typical LaTeX learning curve. I have tried to keep it as simple as possible while still making it powerful enough to handle everything you'll need for a thesis. I hope this

---

[1] These might not be the "best" practices, however, these are practices that I follow to make my work more constant.

document makes your life a little easier and that you find LaTeX as useful as I have.

Disclaimer: This guide provides practical insights and tips for writing a thesis, based on the information available at the time of writing. However, it remains *your responsibility* to verify the accuracy of all content, conclusions, and interpretations. Official university guidelines, requirements, and advice *always* takes precedence over this guide. I assume *no responsibility* for any potential inaccuracies, misrepresentations, or unintended consequences resulting from the use of any material presented in this guide or the supporting documentation and files.

To...

"Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor."

- Author of the Quote

Acknowledgements

While I would love to acknowledge every individual who has influenced this document, I realize that doing so might mean this document never gets finished.

I want to extend my deepest gratitude to my friends and family for their unwavering support throughout all my accomplishments.

A special thank-you goes out to the developers and other LaTeX users who have contributed to the LaTeX community. Without their work, this guide wouldn't exist in its current form.

I am also incredibly thankful to everyone who provided feedback and support along the way, including those new to LaTeX. Your fresh perspectives were invaluable in motivating the transformation from the previous version of this template into a full-fledged class, template, and this newly added guide to writing a thesis, complete with examples of the most requested elements to include.

Table of Contents

List of Symbols

Constants

ϵ_0	Permittivity of Free Space.	$\epsilon_0 = 8.854 \times 10^{-12}$ F/m
$\hbar$	Reduced Planck Constant.	$\hbar = 1.055 \times 10^{-34}$ Js
μ_0	Permeability of Free Space.	$\mu_0 = 4\pi \times 10^{-7}$ H/m
π	Mathematical Constant Pi.	$\pi \approx 3.14159$
R_e	Rankine Number.	$\mathrm{R}_e = \frac{Lv\rho}{\mu}$
c	Speed of light in a vacuum.	$299,792,458\,m/s$
g	Acceleration due to Gravity.	$g = 9.81$ m/s^2
h	Planck constant.	$6.62607015E-34\,Js$
k	Boltzmann Constant.	$k = 1.380649 \times 10^{-23}$ J/K
R	Gas Constant.	$R = 8.314$ J/(mol·K)

Latin

A	Cross-sectional Area.
a	Acceleration
D	Diameter.
d	Distance
E	Young's Modulus
F	Force.

G Shear Modulus.

I Area Moment of Inertia.

K Elastic Constant

L Length.

M Moment.

m Mass

P Pressure.

T Temperature.

T Torque

t Thickness.

V Volume.

v Velocity

Greek

α Primary Angle

δ Deflection.

λ Wavelength.

σ Normal Stress.

τ Shear Stress.

ε Strain

Chapter 1

Introduction

1.1 Background

As a graduate student from the University of Alberta, I am familiar with the challenging task of writing a thesis that adheres to the GPS Minimum Thesis Formatting Requirements. Using a traditional word processor to create a long document filled with equations and figures can be frustrating due to frequent crashes, file corruption, unpredictable formatting changes, and the inability to output a document in the required PDF/A format for submission to GPS.

To overcome these issues, many students turn to LaTeX as an alternative to conventional word processors.

LaTeX allows students and researchers to focus separately on the content and the formatting of their documents. Because the writing is independent of the formatting, documents can be written in lightweight text editors or LaTeX editors, which also facilitate the compilation of the documents. These editors can often save work after every keystroke, and due to the plaintext format, they are less prone to file corruption. Moreover, LaTeX ensures a consistent and professional appearance throughout the document.

1.2 Objectives

The main objectives of this thesis are:

1. To provide a comprehensive guide on writing a thesis using LaTeX.

2. To assist students and researchers in mastering the nuances of LaTeX document preparation.

3. To showcase best practices for structuring and formatting a thesis in LaTeX.

1.3 Scope and Limitations

Although there are existing templates for writing a thesis in LaTeX for the University of Alberta (see **??** for a list of available templates), none seem to provide all the necessary information for creating an outstanding thesis. Most templates apply "band-aid" solutions to existing classes, such as `report` or `book`, offering a customized title page and methods for including prefatory pages. However, these templates often fall short by not providing tips and best practices on how to include the various sections and parts that make up a thesis. They also fail to offer a solid foundation for those who are new to LaTeX. Many of these templates involve extensive patching and fixing, resulting in a large *preamble* section at the beginning of the template that can be confusing to new LaTeX users and add to the already steep learning curve.

Developer	Last Updated	Link to Template Source
Shivam Garg	May 29, 2023	https://github.com/svmgrg/ualberta_thesis_template
Henry Brausen	Feb 11, 2022	https://github.com/henrybrausen/thesis_template
Bernard Llanos	Oct 05, 2019	https://drive.google.com/file/d/1wKS8fu5e6qiVDRt0VUzEtlW8p7uMyz1T/view?usp=sharing
John Bowman	Sep 30, 2019	https://github.com/vectorgraphics/uofathesis
Hongtao Yang & Benjamin Bernard	Sep 28, 2017	https://github.com/adrs0049/ThesisTemplate
GAME & Hongtao Yang	Feb 03, 2016	https://www.ualberta.ca/computing-science/media-library/grad/candidacy-template-tex.tex
Steven Taschuk	Mar 21, 2012	https://github.com/stebulus/ualberta-math-stat-templates/tree/master/thesis
CMENG	Jul 19, 1999	https://sites.ualberta.ca/CMENG/research/new-control/stythes.html

Table 1.1: List of Other Available Templates.

This template, document class, and guide aim to address these shortcomings by providing all the necessary information to create a well-structured thesis, along with examples to assist in formatting your thesis written in LaTeX. To ensure the robustness and ease of maintenance, I developed the class file from the ground up keeping the additional required packages to a minimum. This makes this LaTeX solution easier to maintain, update, and customize to suit different needs from different areas of the University of Alberta. A key goal with this was to reduce the traditionally steep learning curve associated with LaTeX to ensure that anyone could create an outstanding thesis.

While the class file (`ualberta.cls`) deserves its own comprehensive documentation, this document will focus on more specifically on the template file (`ualberta.tex`), as well as the following points:

- Installation and basic usage of LaTeX.

- Document structure and formatting.

- Inclusion of figures and tables.

- Inclusion of plots and graphs.

- Handling mathematical equations.

- Citations and references using BibTeX.

- Use of JabRef—Reference Manager.

- Inclusion of Code and PDF's.

- And more.

This guide does not cover advanced LaTeX programming or extensive customization of document classes. Instead, the class file `ualberta.cls` provides all the major

document and formatting requirements as provided by GPS, while this document offers references on how to include the various elements that might be required in a thesis. This includes all of the explanations of the packages and macros needed to perform the examples.

1.4 Organization of the Thesis

The thesis is organized into several chapters, each addressing a specific aspect of writing a thesis in LaTeX. The breakdown is as follows:

- ??: ??

- ??: ??

- ??: ??

- ??: ??

- ??: ??

- ??: ??

- ??: ??

Each chapter provides detailed information, examples, and recommendations to help you navigate the thesis writing process within the LaTeX ecosystem.

1.5 Summary

This chapter introduced the background, objectives, scope, and organization of the thesis. The subsequent chapters delve into specific topics, offering practical guidance and examples for mastering the art of writing a thesis in LaTeX. Through this process,

you will develop an understanding of how to create a thesis and manipulate content within the LaTeX ecosystem to produce an exceptional document.

Chapter 2

Getting Started with LaTeX

2.1 Installation

To begin using LaTeX, you need to install a LaTeX distribution on your computer. Here there are two main distributors: TeX Live, a free software distribution for the TeX typesetting system which is provided by the TeX User Group (TUG); or MiKTeX, a free and open-source distribution of the TeX/LaTeX typesetting system developed by Christian Schenk. Here are the links for installing LaTeX on different platforms:

2.1.1 Windows

For Windows users, you can install TeX Live or MiKTeX. Download the installer from the respective websites and follow the installation instructions.

2.1.2 Mac

On Mac, you can use MacTeX (Mac version of TeX Live) or MiKTeX. Download the package from the respective websites and follow the installation instructions.

2.1.3 Linux

For Linux users, TeX Live is a common choice or one can use MiKTeX. Use your package manager to install it, or download the installer from the respective websites.

2.2   Basic Document Structure

Once LaTeX is installed, you can create a basic LaTeX document. For a test of the system, open the included program TeXWorks. Once it is open, copy the code below and press the typeset button (green play button). When the document has compiled you will get a preview of the document in a new window. Congratulations, you have compiled a new LaTeX document.

```latex
\documentclass{article}
\begin{document}
    \title{My First \LaTeX{} Document}
    \author{Your Name}
    \date{\today}

    \maketitle

    Hello, \LaTeX{}!
\end{document}
```

This example demonstrates a simple LaTeX document with a title, author, and date. The `\maketitle` is a predefined command that generates the title information. The `Hello, \LaTeX{}!` generate the text: Hello LaTeX! And the `\begin{document}` & `\end{document}` mark the beginning and end of the document's content.

2.3   Other Software Considerations

While by itself LaTeX can be used with just a text editor and compiler, there are some additional software resources that will be very useful.

First of these softwares is a Reference Manager.

Note: *While a reference manager is suited for creating a thesis in LATEX, it is not exclusive for use with LATEX. In fact, even if you are writing your thesis in word I would recommend the use of the suggested reference managers to help keep track of all your references, and to keep track of the information you found within your references. Further, most of these reference managers provide an option to integrate as a word Add-In or allow the export of the bibliography in a word compatible file.*

Second of these is a different LATEX editor. While all of the distributions above come with TeXWorks (a simple editor that allows for compilation) you might find it more beneficial to have some additional features. For this I recommend one of the following:

- TeXstudio

- Texmaker

- TeXnicCenter

- ~~Overleaf~~[1]

I am sure there are others, but any of these will do you fine for writing your thesis.

---

[1]Due to changes with Overleaf, I recommend it less and less. While it is convenient as it is web based, they have restricted it more and more which forces the users to now pay more for the software... including to get longer compile times. While this is not an issue for short articles, in longer documents, such as a thesis, this can be a large hindrance and cost ($100 CAD/year with student discount).

Chapter 3

Discrete-time Control for Isothermal System

3.1 Introduction

Many chemical and petrochemical processes, such as reactions in tubular reactors, heat transfer in exchangers, and separations in columns, involve states distributed in space and time. These systems, known as distributed parameter systems (DPS), are often modeled using partial differential equations (PDEs) to describe distributed state dynamics. Due to their infinite-dimensional nature, the control and estimation of DPSs are inherently more challenging compared to the well-established control theories for finite-dimensional systems [0], making this field an active area of research. Two primary methods, "Early Lumping" and "Late Lumping," have been proposed to address DPS control in the literature. The first, "Early Lumping," reduces the infinite-dimensional system to a finite-dimensional one through spatial discretization during the modeling phase [0]. While this enables standard control strategies, it often compromises model accuracy due to mismatches between the original and reduced-order systems [0]. In contrast, "Late Lumping" preserves the infinite-dimensional system until the final numerical implementation stage, resulting in more accurate but computationally complex control strategies [0].

State reconstruction for DPSs has also been addressed using discrete-time Luenberger observers without spatial discretization, a key feature consistent with the late lumping paradigm [0]. Numerous studies have employed Late Lumping approaches to control infinite-dimensional systems in the field of chemical engineering. These efforts primarily focus on convection-reaction systems governed by first-order hyperbolic PDEs and diffusion-convection-reaction systems governed by second-order parabolic PDEs. For example, robust control of first-order hyperbolic PDEs was explored in [0], where a plug flow reactor system was stabilized under distributed input. Similarly, boundary feedback stabilization using the backstepping method was proposed in [0] for such systems. State feedback regulator design for a countercurrent heat exchanger, another example of a chemical engineering DPS, was addressed in [0]. Introducing the effects of dispersion in tubular reactors, robust control of second-order parabolic PDEs was studied in [0]. Modal decomposition methods for designing low-dimensional predictive controllers for diffusion-convection-reaction systems have also been applied in [0], while observer-based model predictive control (MPC) was developed in [0] for axial dispersion tubular reactors, considering recycle stream effects.

Delay systems represent another class of infinite-dimensional systems studied extensively [0]. Commonly modeled using delay differential equations (DDEs), delays can alternatively be described using transport PDEs, offering advantages in complex scenarios [0]. In chemical engineering DPS control, input/output delays have been widely addressed, as industrial processes often feature both measurement and actuation delays. Such delays are typically handled by modeling them as transportation lag blocks, resulting in cascade PDE systems [0]. State delays, though less common, have been investigated in specific applications, such as heat exchangers with stream delays between passes [0], and plug flow tubular reactors with recycle delays [0]; with the effect of dispersion not being addressed in any of these works. Even in [0], where

the effect of recycle is studied for an axial dispersion tubular reactor, the recycle is assumed to be instantaneous, leaving a gap in the literature regarding state delays in diffusion-convection-reaction systems with recycle streams.

In this work, an axial dispersion reactor with recycle is modeled as a diffusion-convection-reaction DPS. The reactor dynamics are described by a second-order parabolic PDE, coupled with a first-order hyperbolic transport PDE to account for the recycle stream's state delay. A Late Lumping approach is employed, obtaining the system's resolvent in a closed operator form without spatial discretization. To implement MPC as a digital controller, the system is discretized using the Cayley-Tustin method, a Crank-Nicolson-type discretization that conserves the continuous system's characteristics, avoiding the need for model reduction. Numerical simulations demonstrate that the proposed controller stabilizes an unstable system optimally under input constraints. A discrete-time infinite-dimensional Luenberger observer is designed to reconstruct unmeasured states, enabling output feedback MPC. Simulations show that the proposed controller successfully stabilizes the otherwise unstable system under input constraints.

3.2 Mathematical Modeling of the Reactor System

3.2.1 Model representation

The chemical process depicted in Fig. **??** illustrates a chemical reaction within an axial dispersion tubular reactor [0] where reactant A is converted into products. The reactor features a recycle mechanism, allowing a portion of the product stream to re-enter the reactor, ensuring the consumption of any unreacted substrate. The dynamics of the reactant concentration can be described by the second-order parabolic PDE given by (**??**), a common class of equations used to characterize diffusion-convection-reaction systems [0]. The resulting PDE that describes the reactor model is obtained

by utilizing first-principle modeling through relevant mass balance relations on an infinitesimally thin disk element along the longitudinal axis of the reactor.
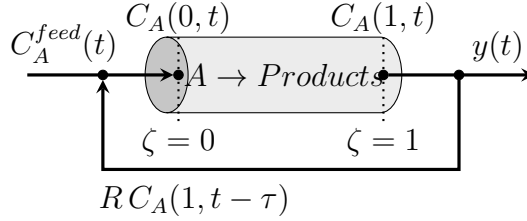

Figure 3.1: Axial tubular reactor with recycle stream.

$$\dot{C}_A(\zeta, t) = D\partial_{\zeta\zeta}C_A(\zeta, t) - v\partial_\zeta C_A(\zeta, t) - r(C_A) \tag{3.1}$$

Here, $C_A(\zeta, t)$ is the concentration of reactant A along the reactor. The physical parameters D and v represent the diffusion coefficient and flow velocity along the reactor, respectively. Physical parameters are assumed to be constant, hence changes in temperature or pressure will not affect the reactor model. The coordinate system in space and time is represented by ζ and t, where $\zeta \in [0, 1]$ and $t \in [0, \infty)$. In addition, $r(C_A)$ is the reaction rate of the reactant in general, which is often a non-linear function of C_A. Therefore, the model is further linearized around its steady-state, followed by introducing the deviation variable $c(\zeta, t) = C_A(\zeta, t) - C_{A,ss}(\zeta)$, where $C_{A,ss}(\zeta)$ is the steady-state concentration of the reactant. The linearized model is then given by (**??**).

$$\dot{c}(\zeta, t) = D\partial_{\zeta\zeta}c(\zeta, t) - v\partial_\zeta c(\zeta, t) - k_r c(\zeta, t) \tag{3.2}$$

Here, $k_r \equiv \left.\dfrac{\partial r(C_A)}{\partial C_A}\right|_{C_{A,ss}}$ is the linearized reaction rate coefficient in the vicinity of the steady-state. The system input is defined as $u(t) \equiv C_A^{feed} - C_{A,ss}^{feed}$, representing the deviation of the concentration of the reactant being fed into the reactor from its steady-state value. The output of the system is also considered as the deviation

of the concentration of the reactant being measured at the reactor outlet from its steady-state value, denoted as $y(t)$.

To accurately represent the behavior of the given axial dispersion tubular reactor, Dankwerts boundary conditions are applied; as they effectively capture deviations from ideal mixing and piston flow while assuming negligible transport lags in connecting lines [0]. The inlet boundary condition is modified to reflect the mixing of the input stream with the delayed state, i.e. the recycled reactant concentration coming from the reactor outlet, occurring τ seconds earlier. These boundary conditions are therefore summarized in (??), with R and τ denoting the recycle ratio and the residence time in the recycle stream, respectively. The system output will consequently be defined as $y(t) = x_1(1, t)$.

$$\begin{cases} D\partial_\zeta c(0, t) - vc(0, t) = -v\left[Rc(1, t - \tau) + (1 - R)u(t)\right] \\ \partial_\zeta c(1, t) = 0 \\ y(t) = c(1, t) \end{cases} \tag{3.3}$$

In the case where the problem involves similar forms of PDEs, an effective general practice to address delays in systems is to reformulate the problem such that the notion of delay is replaced with an alternative transport PDE. Therefore, a new state variable $\underline{x}(\zeta, t) \equiv [x_1(\zeta, t), x_2(\zeta, t)]^T$ is defined as a vector of functions, where $x_1(\zeta, t)$ represents the concentration within the reactor—analogous to $c(\zeta, t)$—and $x_2(\zeta, t)$ is introduced as a new state variable to account for the concentration along the recycle stream. The delay is thus modeled as a pure transport process, wherein the first state $x_1(\zeta, t)$ is transported from the reactor outlet to the inlet, experiencing a delay of τ time units while in the recycle stream. This makes all state variables expressed explicitly at a specific time instance t, resulting in the standard state-space form for

a given infinite-dimensional linear time-invariant (LTI) system given in (**??**).

$$\dot{\underline{x}}(\zeta, t) = \mathfrak{A}\underline{x}(\zeta, t) + \mathfrak{B}u(t)$$

$$y(t) = \mathfrak{C}\underline{x}(\zeta, t) + \mathfrak{D}u(t)$$

(3.4)

Here, $\mathfrak{A}$ is a linear operator $\mathcal{L}(X)$ acting on a Hilbert space $X : L^2[0,1] \times L^2[0,1]$ and $\underline{x}(\zeta, t)$, as defined previously, is the vector of functions describing the states of the system. Input operator $\mathfrak{B}$ is a linear operator that maps the scalar input from input-space onto the state space. Output operator $\mathfrak{C}$ on the other hand, is a linear operator that maps the infinite-dimensional state space onto the finite-dimensional output space, resulting in a scalar output. The operator $\mathfrak{D}$ is the direct transmission operator, which is set to zero in this case as there is no direct transmission from the input to the output in the continuous-time system. The operators ($\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$, and $\mathfrak{D}$) are shown in (**??**) for the infinite-dimensional LTI system.

$$\mathfrak{A} \equiv \begin{bmatrix} D\partial_{\zeta\zeta} - v\partial_\zeta + k_r & 0 \\ 0 & \frac{1}{\tau}\partial_\zeta \end{bmatrix}$$

$$D(\mathfrak{A}) = \Big\{ \underline{x}(\zeta) = [x_1(\zeta), x_2(\zeta)]^T \in X :$$

$$\underline{x}(\zeta), \partial_\zeta \underline{x}(\zeta), \partial_{\zeta\zeta}\underline{x}(\zeta) \quad \text{a.c.,}$$

$$D\partial_\zeta x_1(0) - v x_1(0) = -v R x_2(0),$$

$$\partial_\zeta x_1(1) = 0, x_1(1) = x_2(1) \Big\}$$

$$\mathfrak{B} \equiv \begin{bmatrix} \delta(\zeta) \\ 0 \end{bmatrix} v(1 - R)$$

$$\mathfrak{C} \equiv \begin{bmatrix} \int_0^1 \delta(\zeta - 1)(\cdot)d\zeta & 0 \end{bmatrix}$$

$$\mathfrak{D} = 0$$

(3.5)

where $\delta(\zeta)$ is dirac delta function. This will enable the derivation of the system's spectrum using the eigenvalue problem. The characteristics equation of the system is obtained by solving the equation $det(\mathfrak{A} - \lambda_i I) = 0$ for λ_i, where $\lambda_i \in \mathbb{C}$ is the ith

eigenvalue of the system and I is the identity operator. Attempts to analytically solve this equation have failed; therefore, it is solved numerically using the parameters in Table I. These parameters are carefully chosen to reflect key characteristics of the system, i.e. diffusion, convection, reaction, and delayed recycle.

Similar to [0], a negative reaction coefficient (k_r) is used to induce instability for analysis, a condition uncommon for isothermal reactors but possible in specific cases like autocatalytic or inhibitory reactions. Figure 2 depicts the resulting eigenvalue distribution in the complex plane, confirming instability of the linearized model near its steady state.
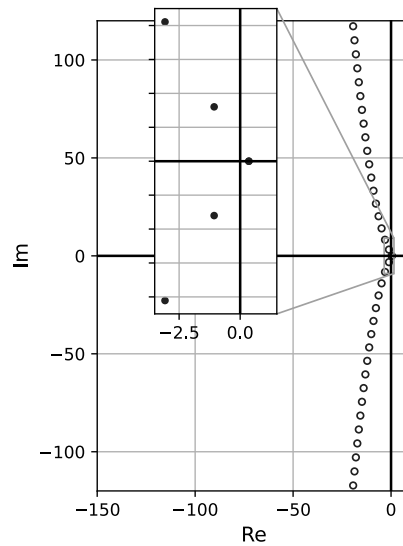
Figure 3.2: Eigenvalues of operator $\mathfrak{A}$.

3.2.2 Adjoint system

Next step is to obtain the adjoint system operators $\mathfrak{A}^*$ and $\mathfrak{B}^*$. Utilizing the relation $\langle \mathfrak{A}\underline{x} + \mathfrak{B}u, \underline{y} \rangle = \langle \underline{x}, \mathfrak{A}^*\underline{y} \rangle + \langle u, \mathfrak{B}^*\underline{y} \rangle$, the adjoint operators $\mathfrak{A}^*$ and $\mathfrak{B}^*$ are obtained as shown in (??) and (??), respectively.

Table 3.1: Physical Parameters for the System

Parameter	Symbol	Value	Unit
Diffusivity	D	2×10^{-5}	m^2/s
Velocity	v	0.01	m/s
Reaction Constant	k_r	-1.5	s^{-1}
Recycle Residence Time	τ	80	s
Recycle Ratio	R	0.3	$-$

$$\mathfrak{A}^* = \begin{bmatrix} D\partial_{\zeta\zeta} + v\partial_\zeta - k_r & 0 \\ 0 & -\frac{1}{\tau}\partial_\zeta \end{bmatrix}$$

$$D(\mathfrak{A}^*) = \left\{ \underline{y} = [y_1, y_2]^T \in Y : \right.$$

$$\underline{y}(\zeta), \partial_\zeta \underline{y}(\zeta), \partial_{\zeta\zeta} \underline{y}(\zeta) \quad \text{a.c.},$$

$$D\partial_\zeta y_1(1) + vy_1(1) = \frac{1}{\tau}y_2(1),$$

$$\left. Rvy_1(0) = \frac{1}{\tau}y_2(0), \partial_\zeta y_1(0) = 0 \right\} \tag{3.6}$$

$$\mathfrak{B}^*(\cdot) = \left[v(1 - R) \int_0^1 \delta(\zeta)(\cdot)d\zeta \quad , \quad 0 \right] \tag{3.7}$$

Once the adjoint operators are determined, the eigenfunctions $\{\underline{\phi_i}(\zeta), \underline{\psi_i}(\zeta)\}$ (for $\mathfrak{A}$ and $\mathfrak{A}^*$, respectively) may be obtained and properly scaled following the calculation of eigenvalues. The set of scaled eigenfunctions will then form a bi-orthonormal basis for the Hilbert space X; which will be later used in the controller design. It is important to note that the system is not self adjoint, as the obtained adjoint operator and its domain are not the same as the original operator and its domain.

3.2.3 Resolvent operator

One must obtain the resolvent operator of the system $\mathfrak{R}(s, \mathfrak{A}) = (sI - \mathfrak{A})^{-1}$ prior to constructing the discrete-time representation of the system. One way to obtain

it is by utilizing the modal characteristics of the system, resulting in an infinite-sum representation of the operator. While being a common practice in the literature, truncating the infinite-sum representation for numerical implementation may lead to a loss of accuracy. Another way to express the resolvent operator is by treating it as an operator that maps either the initial condition of the system $\underline{x}(\zeta, 0)$ or the input $u(t)$, to the Laplace transform of the state of the system $\underline{X}(\zeta, s)$. This approach, although more computationally intensive, results in a closed form expression for the resolvent operator, preserving the infinite-dimensional nature of the system. In (**??**), Laplace transform is applied to the LTI representation of the system for both zero-input response and zero-state response to obtain a general expression for the resolvent operator.

$$\dot{\underline{x}}(\zeta, t) = \mathfrak{A}\underline{x}(\zeta, t) + \mathfrak{B}u(t) \xrightarrow{\mathcal{L}}$$

$$s\underline{X}(\zeta, s) - \underline{x}(\zeta, 0) = \mathfrak{A}\underline{X}(\zeta, s) + \mathfrak{B}U(s)$$

$$\begin{cases} \xrightarrow{u=0} & \underline{X}(\zeta, s) = (sI - \mathfrak{A})^{-1}\underline{x}(\zeta, 0) = \mathfrak{R}(s, \mathfrak{A})\underline{x}(\zeta, 0) \\ \xrightarrow{\underline{x}(0,\zeta)} & \underline{X}(\zeta, s) = (sI - \mathfrak{A})^{-1}\mathfrak{B}U(s) = \mathfrak{R}(s, \mathfrak{A})\mathfrak{B}U(s) \end{cases} \tag{3.8}$$

The goal is to obtain the solution for $\underline{X}(\zeta, s)$ and compare it with the general expression obtained in (**??**) to get the closed form expression for the resolvent operator. First step is to apply Laplace transform to the original system of PDEs in (**??**). The second order derivative term is decomposed to two first order PDEs, constructing a new 3×3 system of first order ODEs with respect to ζ after Laplace transformation, as shown in (**??**).

$$\partial_\zeta \overbrace{\begin{bmatrix} X_1(\zeta, s) \\ \partial_\zeta X_1(\zeta, s) \\ X_2(\zeta, s) \end{bmatrix}}^{\underline{\tilde{X}}(\zeta,s)} = \overbrace{\begin{bmatrix} 0 & 1 & 0 \\ \frac{s+k_r}{D} & \frac{v}{D} & 0 \\ 0 & 0 & s\tau \end{bmatrix}}^{P(s)} \begin{bmatrix} X_1(\zeta, s) \\ \partial_\zeta X_1(\zeta, s) \\ X_2(\zeta, s) \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} 0 \\ -\frac{x_1(\zeta,0)}{D} + v(1 - R)\delta(\zeta)U(s) \\ -\tau x_2(\zeta, 0) \end{bmatrix}}_{Z(\zeta,s)} \tag{3.9}$$

$$\Rightarrow \partial_\zeta \underline{\tilde{X}}(\zeta, s) = P(s)\underline{\tilde{X}}(\zeta, s) + Z(\zeta, s)$$

with solution given by (**??**).

$$\underline{\tilde{X}}(\zeta, s) = \underbrace{e^{P(s)\zeta}}_{T(\zeta,s)} \underline{\tilde{X}}(0, s) + \int_0^\zeta \underbrace{e^{P(s)(\zeta-\eta)}}_{F(\zeta,\eta)} Z(\eta, s)d\eta \tag{3.10}$$

Since the boundary conditions are not homogeneous, $\underline{\tilde{X}}(0, s)$ needs to be obtained by solving the system of algebraic equations given in (**??**); which is the result of applying Danckwerts boundary conditions to the Laplace transformed system of PDEs at $\zeta = 1$.

$$\overbrace{\begin{bmatrix} -v & D & Rv \\ T_{11}(1, s) & T_{12}(1, s) & -T_{33}(1, s) \\ T_{21}(1, s) & T_{22}(1, s) & 0 \end{bmatrix}}^{M^{-1}(s)} \underline{\tilde{X}}(0, s) =$$

$$\underbrace{\int_0^1 \begin{bmatrix} 0 \\ F_{33}(1, \eta)Z_3(\eta, s) - F_{12}(1, \eta)Z_2(\eta, s) \\ -F_{22}(1, \eta)Z_2(\eta, s) \end{bmatrix} d\eta}_{\underline{b}(s)} \tag{3.11}$$

$$\Rightarrow \underline{\tilde{X}}(0, s) = M(s)\underline{b}(s)$$

Having access to $\underline{\tilde{X}}(0, s)$, the solution for $\underline{X}(\zeta, s)$ can be explicitly derived. The

resolvent operator for zero-input and zero-state cases are therefore obtained in a closed form as shown in (**??**) and (**??**), respectively.

$$U(s) = 0 \Rightarrow \mathfrak{R}(s, \mathfrak{A})\underline{(\cdot)} = \begin{bmatrix} \mathfrak{R}_{11} & \mathfrak{R}_{12} \\ \mathfrak{R}_{21} & \mathfrak{R}_{22} \end{bmatrix} \begin{bmatrix} (\cdot)_1 \\ (\cdot)_2 \end{bmatrix} \Rightarrow$$

$$\mathfrak{R}_{11} = \sum_{j=1}^{2} \frac{T_{1j}(\zeta)}{D} \int_{0}^{1} [M_{j2}F_{12}(1, \eta) + M_{j3}F_{22}(1, \eta)] (\cdot)_1 d\eta$$

$$- \frac{1}{D} \int_{0}^{\zeta} F_{12}(\zeta, \eta)(\cdot)_1 d\eta$$

$$\mathfrak{R}_{12} = \sum_{j=1}^{2} -\tau T_{1j}(\zeta) \int_{0}^{1} M_{j2}F_{33}(1, \eta)(\cdot)_2 d\eta \tag{3.12}$$

$$\mathfrak{R}_{21} = \frac{T_{33}(\zeta)}{D} \int_{0}^{1} [M_{32}F_{12}(1, \eta) + M_{33}F_{22}(1, \eta)] (\cdot)_1 d\eta$$

$$\mathfrak{R}_{22} = -\tau T_{33}(\zeta) \int_{0}^{1} M_{32}F_{33}(1, \eta)(\cdot)_2 d\eta$$

$$- \tau \int_{0}^{\zeta} F_{33}(\zeta, \eta)(\cdot)_2 d\eta$$

$$\underline{x}(\zeta, 0) = 0 \Rightarrow \mathfrak{R}(s, \mathfrak{A})\mathfrak{B}(\cdot) = \begin{bmatrix} \mathfrak{R}_1\mathfrak{B} \\ \mathfrak{R}_2\mathfrak{B} \end{bmatrix} (\cdot) \Rightarrow$$

$$\mathfrak{R}_1\mathfrak{B} = -\upsilon(1 - R)\Big[\sum_{j=1}^{2} T_{1j}(\zeta)(M_{j2}T_{12}(1) + M_{j3}T_{22}(1)) \tag{3.13}$$

$$- T_{12}(\zeta)\Big](\cdot)$$

$$\mathfrak{R}_2\mathfrak{B} = -\upsilon(1 - R)\left[T_{33}(\zeta)(M_{32}T_{12}(1) + M_{33}T_{22}(1))\right](\cdot)$$

Since the system generator $\mathfrak{A}$ is not self-adjoint, the resolvent operator for the adjoint system shall also be obtained. This is done in a similar manner as the original system, resulting in a closed-form expression for the adjoint resolvent operator $\mathfrak{R}^*(s, \mathfrak{A}^*)$. To avoid redundancy, the derivation of the resolvent operator for the adjoint system is not included in this manuscript.

3.2.4   Cayley–Tustin Time Discretization

To implement the system on digital controllers, it is necessary to transition to a discrete-time framework while preserving critical properties such as stability and controllability. The Cayley–Tustin time-discretization method achieves this by mapping the continuous-time system to the discrete domain [0]. This Crank–Nicolson type of discretization is also known as the lowest order symplectic integrator in Gauss quadrature-based Runge–Kutta methods [0]. Considering Δt as the sampling time, and assuming a piecewise constant input within time intervals (zero-order hold), the discrete-time representation $\underline{x}(\zeta, k) = \mathfrak{A}_d \underline{x}(\zeta, k-1) + \mathfrak{B}_d u(k)$ is obtained, with discrete-time operators $\mathfrak{A}_d$, $\mathfrak{B}_d$, $\mathfrak{C}_d$, and $\mathfrak{D}_d$ defined in (??), where $\alpha = 2/\Delta t$.

$$\begin{bmatrix} \mathfrak{A}_d & \mathfrak{B}_d \\ \mathfrak{C}_d & \mathfrak{D}_d \end{bmatrix} = \begin{bmatrix} -I + 2\alpha \mathfrak{R}(\alpha, \mathfrak{A}) & \sqrt{2\alpha} \mathfrak{R}(\alpha, \mathfrak{A})\mathfrak{B} \\ \sqrt{2\alpha} \mathfrak{C}\mathfrak{R}(\alpha, \mathfrak{A}) & \mathfrak{C}\mathfrak{R}(\alpha, \mathfrak{A})\mathfrak{B} \end{bmatrix} \tag{3.14}$$

As required for systems with nonself-adjoint generators, the adjoint discrete-time operators $\mathfrak{A}_d^*$ and $\mathfrak{B}_d^*$ are also obtained in a similar manner, as shown in (??).

$$\begin{bmatrix} \mathfrak{A}_d^* & \mathfrak{B}_d^* \end{bmatrix} = \begin{bmatrix} -I + 2\alpha \mathfrak{R}^*(\alpha, \mathfrak{A}^*) & \sqrt{2\alpha} \mathfrak{B}^* \mathfrak{R}^*(\alpha, \mathfrak{A}^*) \end{bmatrix} \tag{3.15}$$

3.3   Estimation and Control

3.3.1   Model predictive control design, full-state availability

The proposed full-state feedback model predictive control strategy, as shown in Fig. ??, is developed in this section with the goal of stabilizing the given unstable infinite-dimensional system within an optimal framework while satisfying input constraints. An infinite-time open-loop objective function sets the foundation of the controller design in the discrete-time setting at each sampling instant k, which consists of a weighted sum of state deviations and actuation costs for all future time instances,

subject to the system dynamics and input constraints, as shown in (**??**).
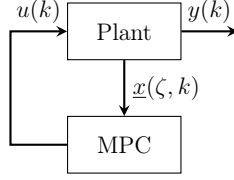
Figure 3.3: Proposed full-state feedback model predictive control system.

$$\min_{U} \quad \sum_{l=0}^{\infty} \langle \underline{x}(\zeta, k+l|k), \mathfrak{Q}\underline{x}(\zeta, k+l|k) \rangle$$
$$+ \langle u(k+l+1|k), \mathfrak{F}u(k+l+1|k) \rangle$$

$$(3.16)$$

$$\text{s.t.} \quad \underline{x}(\zeta, k+l|k) = \mathfrak{A}_d\underline{x}(\zeta, k+l-1|k) + \mathfrak{B}_d u(k+l|k)$$

$$u^{min} \leq u(k+l|k) \leq u^{max}$$

where $\mathfrak{Q}$ and $\mathfrak{F}$ are positive definite operators of appropriate dimensions, responsible for penalizing state deviations and actuation costs, respectively. The notation $(k+l|k)$ indicates the future time states or input instance $k+l$ obtained at time k. The infinite-time optimization problem may be reduced to a finite-time setup by assigning zero-input beyond a certain control horizon N, resulting in the optimization problem in (**??**).

$$\min_{U} \quad \sum_{l=0}^{N-1} \langle \underline{x}(\zeta, k+l|k), \mathfrak{Q}\underline{x}(\zeta, k+l|k) \rangle$$
$$+ \langle u(k+l+1|k), \mathfrak{F}u(k+l+1|k) \rangle$$
$$+ \langle \underline{x}(\zeta, k+N|k), \mathfrak{P}\underline{x}(\zeta, k+N|k) \rangle$$

$$(3.17)$$

$$\text{s.t.} \quad \underline{x}(\zeta, k+l|k) = \mathfrak{A}_d\underline{x}(\zeta, k+l-1|k) + \mathfrak{B}_d u(k+l|k)$$

$$u^{min} \le u(k+l|k) \le u^{max}$$

$$\langle \underline{x}(\zeta, k+N|k), \underline{\phi_u}(\zeta) \rangle = 0$$

Obtained as the solution to the discrete-time Lyapunov equation, $\mathfrak{P}$ is the terminal cost operator as shown in (**??**); which can be proven to be positive definite only if the terminal state $\underline{x}(\zeta, k+N|k)$ is in a stable subspace. Therefore, an equality constraint is introduced to guarantee that the resulting quadratic optimization problem is convex. The terminal constraint is enforced by setting the projection of the terminal state onto the unstable subspace of the system to zero [0]. Here, $\underline{\phi_u}(\zeta)$ is the set of unstable eigenfunctions of the system, for all eigenvalues where $\text{Re}(\lambda_u) \ge 0$.

$$\mathfrak{P}(\cdot) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} -\frac{\langle \phi_m, \mathfrak{Q}\psi_n \rangle}{\lambda_m + \overline{\lambda_n}} \langle (\cdot), \underline{\psi_n} \rangle \phi_m \qquad (3.18)$$

One may further process the optimization problem in (**??**) to obtain a standard format for quadratic programming (QP) solvers by substituting the future states in terms of the current state and the sequence of future inputs using system dynamics expression. The resulting QP problem is given in (**??**). The optimal input sequence U is then obtained by solving the QP problem at each sampling instant k. To implement a receding horizon control strategy, only the first input of the optimal sequence $u(k+1|k)$ is applied to the system, and the optimization problem is solved again at the next sampling instant $k+1$.

$$\min_U J = U^T \langle I, H \rangle U + 2U^T \langle I, P\underline{x}(\zeta, k|k) \rangle$$

$$\text{s.t.} \quad U^{min} \leq U \leq U^{max}$$

$$T_u \underline{x}(\zeta, k|k) + S_u U = 0$$

with $H =$

$$
\begin{bmatrix}
\mathfrak{B}_d^* \mathfrak{P} \mathfrak{B}_d + \mathfrak{F} & \mathfrak{B}_d^* \mathfrak{A}_d^* \mathfrak{P} \mathfrak{B}_d & \cdots & \mathfrak{B}_d^* \mathfrak{A}_d^{*N-1} \mathfrak{P} \mathfrak{B}_d \\
\mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d \mathfrak{B}_d & \mathfrak{B}_d^* \mathfrak{P} \mathfrak{B}_d + \mathfrak{F} & \cdots & \mathfrak{B}_d^* \mathfrak{A}_d^{*N-2} \mathfrak{P} \mathfrak{B}_d \\
\vdots & \vdots & \ddots & \vdots \\
\mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d^{N-1} \mathfrak{B}_d & \mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d^{N-2} \mathfrak{B}_d & \cdots & \mathfrak{B}_d^* \mathfrak{P} \mathfrak{B}_d + \mathfrak{F}
\end{bmatrix} \tag{3.19}
$$

$$P = \begin{bmatrix} \mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d & \mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d^2 & \dots & \mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d^N \end{bmatrix}^T$$

$$T_u(\cdot) = \begin{bmatrix} \langle \mathfrak{A}_d^N(\cdot), \underline{\phi_u} \rangle \end{bmatrix}$$

$$S_u = \begin{bmatrix} \langle \mathfrak{A}_d^{N-1} \mathfrak{B}_d, \underline{\phi_u} \rangle & \langle \mathfrak{A}_d^{N-2} \mathfrak{B}_d, \underline{\phi_u} \rangle & \dots & \langle \mathfrak{B}_d, \underline{\phi_u} \rangle \end{bmatrix}$$

$$U = \begin{bmatrix} u(k+1|k) & u(k+2|k) & \dots & u(k+N|k) \end{bmatrix}^T$$

3.3.2 Continuous-Time Observer Design

For the purpose of state reconstruction of a diffusion-convection-reaction system, where the feedforward term $\mathfrak{D}$ is generally absent, the continuous-time observer dynamics are given by (**??**).

$$\dot{\hat{\underline{x}}}(\zeta, t) = \mathfrak{A}\hat{\underline{x}}(\zeta, t) + \mathfrak{B}u(t) + \mathfrak{L}_c[y(t) - \hat{y}(t)]$$

$$\hat{y}(t) = \mathfrak{C}\hat{\underline{x}}(\zeta, t) \tag{3.20}$$

where $\hat{\underline{x}}(\zeta, t)$ is the reconstructed state of the original system and $\mathfrak{L}_c$ is the continuous-time observer gain. By subtracting the observer dynamics from the original system dynamics, the error dynamics $e(\zeta, t)$ are obtained as shown in (**??**).

$$\dot{e}(\zeta, t) = (\mathfrak{A} - \mathfrak{L}_c \mathfrak{C})e(\zeta, t) \equiv \mathfrak{A}_o e(\zeta, t) \tag{3.21}$$

The goal is to design the observer gain $\mathfrak{L}_c$ such that the error dynamics are exponentially stable, i.e. $\max\{\text{Re}(\lambda_o)\} < 0$ where $\{\lambda_o\}$ is the set of eigenvalues of the

error dynamics operator $\mathfrak{A}_o$. Three different forms of the observer gain are considered as spatial functions $\mathfrak{L}_c = f(\zeta, l_{obs})$ with the effect of the scalar coefficient l_{obs} on $\max\{\mathrm{Re}(\lambda_o)\}$ shown in Fig. **??**.
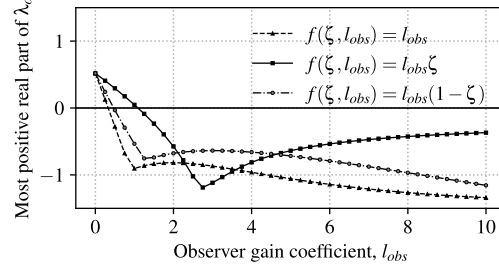

Figure 3.4: The effect of various observer gains $\mathfrak{L}_c = f(\zeta, l_{obs})$ on the eigenvalues of state reconstruction error dynamics λ_o.

3.3.3 Discrete-Time Observer Design

Once an appropriate continuous-time observer gain is determined, the discrete-time observer gain $\mathfrak{L}_d$ may be obtained using the same Cayley-Tustin time discretization approach, as shown in (**??**).

$$\hat{\underline{x}}(\zeta, k) = \mathfrak{A}_d \hat{\underline{x}}(\zeta, k-1) + \mathfrak{B}_d u(k) + \mathfrak{L}_d[y(k) - \hat{y}(k)]$$

$$\hat{y}(k) = \mathfrak{C}_{d,o}\hat{\underline{x}}(\zeta, k-1) + \mathfrak{D}_{d,o}u(k) + \mathfrak{M}_{d,o}y(k)$$

(3.22)

with $\mathfrak{A}_d$ and $\mathfrak{B}_d$ defined in (**??**), and $\mathfrak{C}_{d,o}$, $\mathfrak{D}_{d,o}$, $\mathfrak{M}_{d,o}$, and $\mathfrak{L}_d$ are given in (**??**).

$$\mathfrak{C}_{d,o}(\cdot) = \sqrt{2\alpha}\left[I + \mathfrak{C}(\alpha I - \mathfrak{A})\mathfrak{L}_c\right]^{-1}\mathfrak{C}\mathfrak{R}(\alpha, \mathfrak{A})(\cdot)$$

$$\mathfrak{D}_{d,o} = \left[I + \mathfrak{C}(\alpha I - \mathfrak{A})\mathfrak{L}_c\right]^{-1}\mathfrak{C}\mathfrak{R}(\alpha, \mathfrak{A})\mathfrak{B}$$

$$\mathfrak{M}_{d,o} = \left[I + \mathfrak{C}\mathfrak{R}(\alpha, \mathfrak{A})\mathfrak{L}_c\right]^{-1}\mathfrak{C}\mathfrak{R}(\alpha, \mathfrak{A})\mathfrak{L}_c$$

$$\mathfrak{L}_d = \sqrt{2\alpha}\mathfrak{R}(\alpha, \mathfrak{A})\mathfrak{L}_c$$

(3.23)

It can been shown that using this approach, the discrete-time error dynamics will be stable if the continuous-time observer gain $\mathfrak{L}_c$ is chosen such that $\mathfrak{A}_o$ is stable. It is also worth noting that the proposed methodology skips the need for model reduction associated with the discrete-time Luenberger observer, with no spatial approximation required as well [0].

3.3.4 Model predictive control design, output feedback implementation

To enable real-time implementation under limited state access, the discrete-time model predictive controller is now augmented with the obtained discrete-time Luenberger observer. The reconstructed state $\hat{\underline{x}}(\zeta, k)$ is substituted for the full state in the MPC formulation, yielding an observer-based output-feedback controller, as illustrated in Fig. **??**.
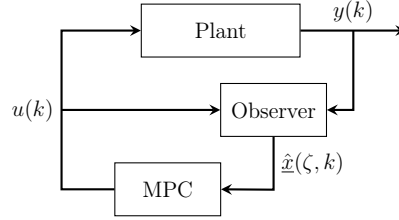
Figure 3.5: Block diagram representation of the observer-based MPC.

The cost function and terminal condition remain unchanged, but the predicted state trajectory is now driven by the estimated state:

$$
\min_{U} \quad \sum_{l=0}^{N-1} \langle \hat{\underline{x}}(\zeta, k+l|k), \mathfrak{Q}\hat{\underline{x}}(\zeta, k+l|k) \rangle
$$
$$
+ \langle u(k+l+1|k), \mathfrak{F}u(k+l+1|k) \rangle
$$
$$
+ \langle \hat{\underline{x}}(\zeta, k+N|k), \mathfrak{P}\hat{\underline{x}}(\zeta, k+N|k) \rangle
$$

$$(3.24)$$

$$
\text{s.t.} \quad \hat{\underline{x}}(\zeta, k+l|k) = \mathfrak{A}_d \hat{\underline{x}}(\zeta, k+l-1|k) + \mathfrak{B}_d u(k+l|k)
$$
$$
u^{min} \le u(k+l|k) \le u^{max}
$$
$$
\langle \hat{\underline{x}}(\zeta, k+N|k), \underline{\phi}_u(\zeta) \rangle = 0
$$

The observer provides $\hat{\underline{x}}(\zeta, k)$ at each time step by processing most recent output and control input. This reconstructed state initializes the prediction horizon and closes the loop in the absence of full-state access. The resulting control law inherits

all properties of the full-state MPC while enabling output feedback implementation. The QP formulation follows analogously by substituting $\underline{x}$ with $\hat{\underline{x}}$ in (**??**), and is detailed in (**??**).

$$\min_{U} J = U^\top \langle I, H \rangle U + 2U^\top \langle I, P\hat{\underline{x}}(\zeta, k|k) \rangle$$

$$\text{s.t.} \quad U^{min} \le U \le U^{max}$$

$$T_u \hat{\underline{x}}(\zeta, k|k) + S_u U = 0$$

with $H =$

$$\begin{bmatrix} \mathfrak{B}_d^* \mathfrak{P} \mathfrak{B}_d + \mathfrak{F} & \mathfrak{B}_d^* \mathfrak{A}_d^* \mathfrak{P} \mathfrak{B}_d & \cdots & \mathfrak{B}_d^* \mathfrak{A}_d^{*N-1} \mathfrak{P} \mathfrak{B}_d \\ \mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d \mathfrak{B}_d & \mathfrak{B}_d^* \mathfrak{P} \mathfrak{B}_d + \mathfrak{F} & \cdots & \mathfrak{B}_d^* \mathfrak{A}_d^{*N-2} \mathfrak{P} \mathfrak{B}_d \\ \vdots & \vdots & \ddots & \vdots \\ \mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d^{N-1} \mathfrak{B}_d & \mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d^{N-2} \mathfrak{B}_d & \cdots & \mathfrak{B}_d^* \mathfrak{P} \mathfrak{B}_d + \mathfrak{F} \end{bmatrix} \tag{3.25}$$

$$P = \begin{bmatrix} \mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d & \mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d^2 & \cdots & \mathfrak{B}_d^* \mathfrak{P} \mathfrak{A}_d^N \end{bmatrix}^\top$$

$$T_u(\cdot) = \begin{bmatrix} \langle \mathfrak{A}_d^N(\cdot), \underline{\phi_u} \rangle \end{bmatrix}$$

$$S_u = \begin{bmatrix} \langle \mathfrak{A}_d^{N-1} \mathfrak{B}_d, \underline{\phi_u} \rangle & \langle \mathfrak{A}_d^{N-2} \mathfrak{B}_d, \underline{\phi_u} \rangle & \cdots & \langle \mathfrak{B}_d, \underline{\phi_u} \rangle \end{bmatrix}$$

$$U = \begin{bmatrix} u(k+1|k) & u(k+2|k) & \cdots & u(k+N|k) \end{bmatrix}^\top$$

3.4   Simulation Results

This section presents numerical simulations of the closed-loop system under both full-state feedback and output-feedback model predictive control schemes. The reactor model and all physical parameters follow those in Table **??**, and the same control settings are used throughout: initial condition $c(\zeta, 0) = \sin^2(\pi\zeta)$, empty recycle stream, state and input penalty weights $Q = 0.04I$, $F = 27$, sampling time $\Delta t = 20$ s, control horizon $N = 9$, and input constraints $0 \le u(t) \le 0.15$. The control horizon corresponds to 180 s, which exceeds the recycle delay of 80 s. The subsections below compare the controller performance under full-state and output-feedback implemen-

tations.

3.4.1 Full-State Feedback MPC Performance

As the eigenvalue distribution obtained in Fig. **??** suggests, the open-loop system is unstable due to the presence of an eigenvalue with positive real part. The zero-input response of the system is shown in Fig. **??** where the initial condition for the reactor is set to $c(\zeta, 0) = \sin^2(\pi\zeta)$. The recycle stream is assumed to be empty at the beginning of the simulation.

An infinite-dimensional MPC is designed and applied to the unstable system. The closed-loop response of the system is shown in Fig. **??** and the control input as well as the measured output is shown in Fig. **??**. It may be confirmed that the MPC successfully stabilizes the unstable system while satisfying the input constraints.
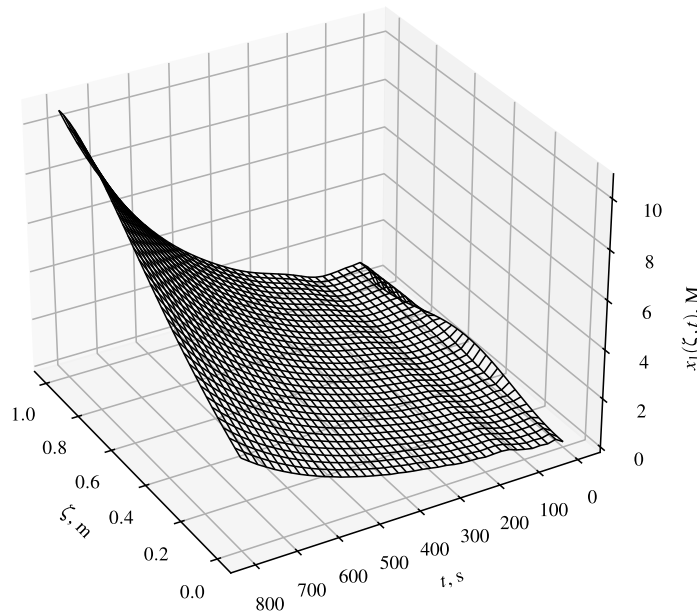
Figure 3.6: Open-loop concentration profile along the reactor.

One interesting aspect of considering a recycle stream is the oscillatory behavior
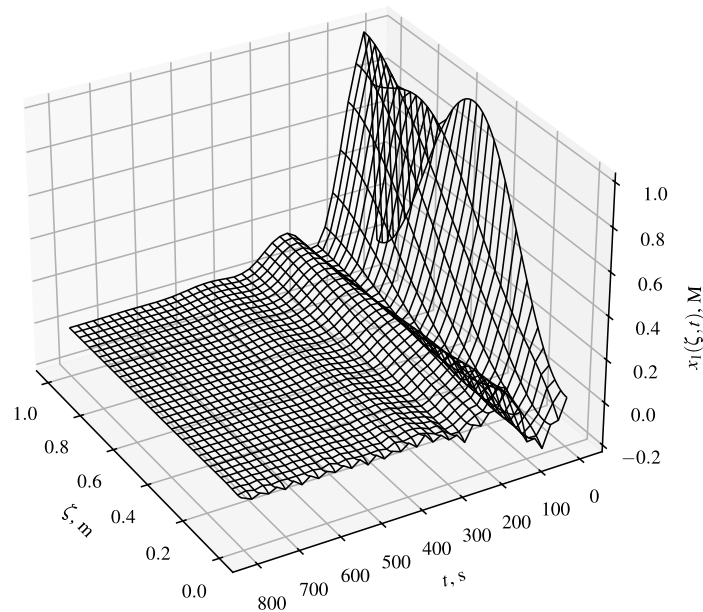
Figure 3.7: Stabilized reactor concentration profile under the proposed full-state MPC.
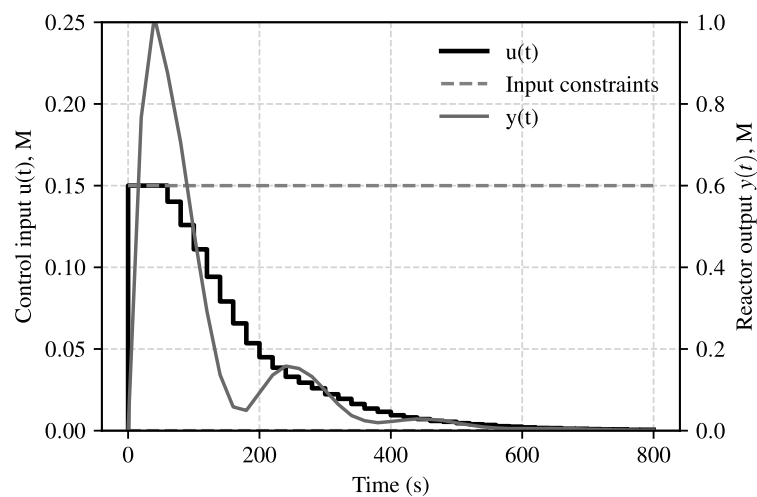
Figure 3.8: Input profile and reactor output under full-state MPC, subject to constraints.

of the system dynamics. While axial dispersion reactors show no oscillation in the absence of recycle, the nature of recycle streams can introduce such behavior. The choice of control horizon is another key factor. A short control horizon relative to the residence time of the recycle stream can lead to oscillatory input profiles due to the presence of delayed recycle stream. In this example, the control horizon, i.e., 180 s, is set to be considerably longer than the recycle delay, which is 80 s; resulting in a non-oscillatory input profile.

3.4.2   Observer-Based Output Feedback MPC

To evaluate the performance of the output-feedback controller, numerical simulations are conducted under the same conditions as in the full-state feedback case. This subsection presents the closed-loop behavior of the system when using the discrete-time Luenberger observer to reconstruct the states based on output measurements.

The eigenvalue distribution shown previously in Fig. **??** confirms that the open-loop system is unstable due to the presence of an eigenvalue with a positive real part. The observer gain is selected as a constant function $L_c = 1$, and the estimated state is initialized to zero across the domain.

The closed-loop reactor response under the proposed output-feedback controller is shown in Fig. **??**, and the corresponding control input and measured output are shown in Fig. **??**. The evolution of the state estimation error is depicted in Fig. **??**. These results confirm that the observer-based MPC successfully stabilizes the unstable system while adhering to input constraints, using only output measurements.

An important aspect of the proposed observer-based controller is the relative speed of the observer error convergence compared to the system dynamics. As seen in Fig. **??**, the observer error dynamics decay significantly faster than the closed-loop reactor response, helping prevent oscillations that may arise from poor state recon-
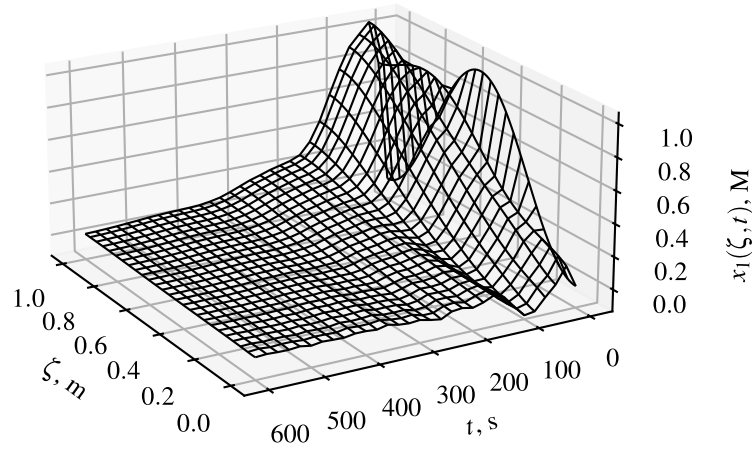
Figure 3.9: Stabilized reactor concentration profile under the proposed observer-based MPC.
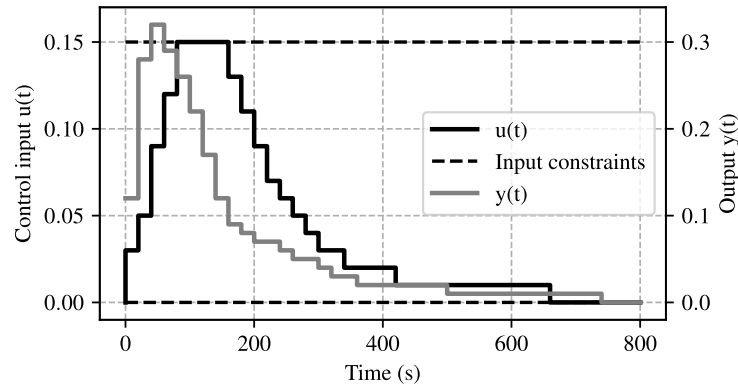
Figure 3.10: Input profile and reactor output under observer-based MPC.
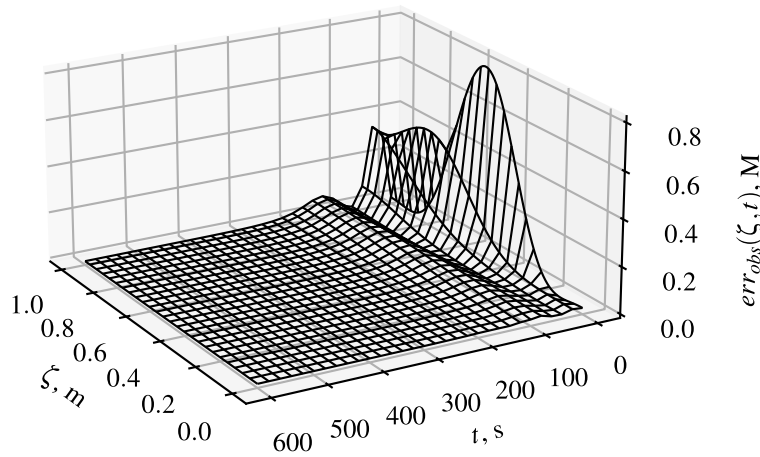
Figure 3.11: State reconstruction error profile along the reactor.

struction.

Oscillatory behavior induced by the recycle stream is discussed at the end of the previous subsection. Since all simulation settings are shared, the same rationale applies here; the long control horizon relative to the recycle delay ensures a smooth input profile and stable closed-loop response.

3.5 Conclusion

In this work, model predictive control of an axial dispersion tubular reactor equipped with recycle is addressed, while considering the delay imposed by the recycle stream. This setup is common in industry but has received limited attention in the chemical engineering distributed parameter systems literature. The diffusion-convection-reaction dynamics of the reactor is modeled by a second-order parabolic PDE, while a notion of state delay is introduced to account for the delay imposed by the recycle stream. The state delay is addressed as a separate transport PDE, resulting in a boundary-controlled system governed by a coupled set of parabolic and hyperbolic PDEs under Danckwerts boundary conditions. Utilizing a late-lumping approach, the resolvent operator is obtained in a closed form in order to preserve the infinite-dimensional nature of the system without requiring spatial discretization. To implement MPC as a digital controller, the Cayley–Tustin transformation is used. This Crank–Nicolson type of discretization is chosen as it maintains important properties of the system such as stability and controllability when mapping the continuous-time system to a discrete-time one. Numerical simulations demonstrate the effectiveness of the proposed controller in stabilizing an unstable system while satisfying input constraints under full-state feedback. Recognizing, however, that full-state information is often unavailable in practical implementations of distributed parameter systems, this work is further extended by designing and integrating a discrete-time Luenberger

observer to reconstruct the state from output measurements alone, without any spatial approximation. A family of observer gains is examined, and spectral analysis is performed to select gains that ensure the state reconstruction error converges faster than the closed-loop system dynamics. This guarantees accurate state estimates during transients and prevents performance degradation due to estimation delay. The proposed approach can be further extended to incorporate the effects of temperature as well as disturbance rejection or set-point tracking in future work.

References

[0]   W. H. Ray, *Advanced process control.* McGraw-Hill: New York, NY, USA, 1981.

[0]   E. Davison, "The robust control of a servomechanism problem for linear time-invariant multivariable systems," *IEEE transactions on Automatic Control*, vol. 21, no. 1, pp. 25–34, 1976, ISSN: 0018-9286. DOI: 10.1109/tac.1976.1101137.

[0]   A. A. Moghadam, I. Aksikas, S. Dubljevic, and J. F. Forbes, "Infinite-dimensional LQ optimal control of a dimethyl ether (DME) catalytic distillation column," *Journal of Process Control*, vol. 22, no. 9, pp. 1655–1669, 2012, ISSN: 0959-1524. DOI: 10.1016/j.jprocont.2012.06.018.

[0]   D. Dochain, "State observers for tubular reactors with unknown kinetics," *Journal of process control*, vol. 10, no. 2-3, pp. 259–268, 2000, ISSN: 0959-1524. DOI: 10.1016/s0959-1524(99)00020-7.

[0]   D. Dochain, "State observation and adaptive linearizing control for distributed parameter (bio) chemical reactors," *International Journal of Adaptive Control and Signal Processing*, vol. 15, no. 6, pp. 633–653, 2001, ISSN: 0890-6327. DOI: 10.1002/acs.691.

[0]   A. A. Alonso, I. G. Kevrekidis, J. R. Banga, and C. E. Frouzakis, "Optimal sensor location and reduced order observer design for distributed process systems," *Computers & chemical engineering*, vol. 28, no. 1-2, pp. 27–35, 2004, ISSN: 0098-1354. DOI: 10.1016/s0098-1354(03)00175-3.

[0]   J. M. Ali, N. H. Hoang, M. A. Hussain, and D. Dochain, "Review and classification of recent observers applied in chemical process systems," *Computers & Chemical Engineering*, vol. 76, pp. 27–41, 2015, ISSN: 0098-1354. DOI: 10.1016/j.compchemeng.2015.01.019.

[0]   P. D. Christofides and P. Daoutidis, "Feedback control of hyperbolic PDE systems," *AIChE Journal*, vol. 42, no. 11, pp. 3063–3086, 1996, ISSN: 0001-1541. DOI: 10.1002/aic.690421108.

[0] M. Krstic and A. Smyshlyaev, "Backstepping boundary control for first-order hyperbolic PDEs and application to systems with actuator and sensor delays," *Systems & Control Letters*, vol. 57, no. 9, pp. 750–758, 2008, ISSN: 0167-6911. DOI: 10.1016/j.sysconle.2008.02.005.

[0] X. Xu and S. Dubljevic, "The state feedback servo-regulator for countercurrent heat-exchanger system modelled by system of hyperbolic PDEs," *European Journal of Control*, vol. 29, pp. 51–61, 2016, ISSN: 0947-3580. DOI: 10.1016/j.ejcon.2016.02.002.

[0] P. D. Christofides, "Robust control of parabolic PDE systems," *Chemical Engineering Science*, vol. 53, no. 16, pp. 2949–2965, 1998, ISSN: 2324-9749. DOI: 10.1007/978-1-4612-0185-4_5.

[0] S. Dubljevic and P. D. Christofides, "Predictive control of parabolic PDEs with boundary control actuation," *Chemical Engineering Science*, vol. 61, no. 18, pp. 6239–6248, 2006, ISSN: 0009-2509. DOI: 10.1016/j.ces.2006.05.041.

[0] S. Khatibi, G. O. Cassol, and S. Dubljevic, "Model predictive control of a non-isothermal axial dispersion tubular reactor with recycle," *Computers & Chemical Engineering*, vol. 145, p. 107 159, 2021, ISSN: 0098-1354. DOI: 10.1016/j.compchemeng.2020.107159.

[0] R. Curtain and H. Zwart. Springer Nature, 2020. DOI: 10.1007/978-1-0716-0590-5_1.

[0] M. Krstić (Systems & control). Birkhäuser, 2009.

[0] S. Hiratsuka and A. Ichikawa, "Optimal control of systems with transportation lags," *IEEE Transactions on Automatic Control*, vol. 14, no. 3, pp. 237–247, 1969, ISSN: 0018-9286. DOI: 10.1109/TAC.1969.1099175.

[0] L. Mohammadi, I. Aksikas, S. Dubljevic, and J. F. Forbes, "LQ-boundary control of a diffusion-convection-reaction system," *International Journal of Control*, vol. 85, no. 2, pp. 171–181, 2012, ISSN: 0959-1524. DOI: 10.1080/00207179.2011.642308.

[0] G. O. Cassol and S. Dubljevic, "Discrete output regulator design for a mono-tubular reactor with recycle," in *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 1262–1267. DOI: 10.23919/acc.2019.8815250.

[0] G. O. Cassol, D. Ni, and S. Dubljevic, "Heat exchanger system boundary regulation," *AIChE Journal*, vol. 65, no. 8, e16623, 2019, ISSN: 0001-1541. DOI: 10.1002/aic.16623.

[0] J. Qi, S. Dubljevic, and W. Kong, "Output feedback compensation to state and measurement delays for a first-order hyperbolic PIDE with recycle," *Automatica*, vol. 128, p. 109 565, 2021, ISSN: 0005-1098. DOI: 10.1016/j.automatica.2021.109565.

[0]   O. Levenspiel, *Chemical reaction engineering.* John Wiley & Sons, 1998.

[0]   K. F. Jensen and W. H. Ray, "The bifurcation behavior of tubular reactors," *Chemical Engineering Science*, vol. 37, no. 2, pp. 199–222, 1982, ISSN: 0009-2509. DOI: 10.1016/0009-2509(82)80155-3.

[0]   P. V. Danckwerts, "Continuous flow systems: Distribution of residence times," *Chemical Engineering Science*, vol. 2, no. 1, pp. 1–13, 1953, ISSN: 0009-2509. DOI: 10.1016/0009-2509(53)80001-1.

[0]   B. Moadeli, G. O. Cassol, and S. Dubljevic, "Optimal control of axial dispersion tubular reactors with recycle: Addressing state-delay through transport PDEs," *The Canadian Journal of Chemical Engineering*, vol. 103, no. 8, pp. 3751–3766, 2025, ISSN: 0008-4034. DOI: 10.1002/cjce.25629.

[0]   V. Havu and J. Malinen, "The Cayley transform as a time discretization scheme," *Numerical Functional Analysis and Optimization*, vol. 28, no. 7-8, pp. 825–851, 2007, ISSN: 0163-0563. DOI: 10.1080/01630560701493321.

[0]   Q. Xu and S. Dubljevic, "Linear model predictive control for transport-reaction processes," *AIChE Journal*, vol. 63, no. 7, pp. 2644–2659, 2017, ISSN: 0001-1541. DOI: 10.1002/aic.15592.

[0]   E. Hairer, M. Hochbruck, A. Iserles, and C. Lubich, "Geometric numerical integration," *Oberwolfach Reports*, vol. 3, no. 1, pp. 805–882, 2006, ISSN: 1660-8933. DOI: 10.4171/owr/2011/16.

Chapter 4

Figures, Tables, & Plates

4.1 Introduction

Figures and Tables play a crucial role in conveying information effectively in academic documents. This chapter will delve into the intricacies of incorporating figures and tables in your LaTeX document, exploring various features and advanced techniques to enhance the visual appeal and clarity of your content. While there are many types of figures that one might have in their document, this Chapter more specifically focuses on the inclusion of graphic figures (pictures), as well as how to lay out multi figures (sub-figures).

⚠ WARNING **Throughout this Chapter there will be code listings that include lines that show the required packages. These lines should be included in your LaTeX preamble, not in the body of your document. To make this easy I have actually included a document specifically to add your packages too. This can be found in the includePackages.tex file found in the 00_LaTeX_Files folder.**

💡 `./00_LaTeX_Files/includePackages.tex` includes all the packages required for creating all the examples in this document, some of these will not be necessary for your own thesis, however, I have included comments on all packages for what they are used for. This allows you to make a decision on if you will need them as well. If you are unsure, you may just comment out the line so as to not forget the packages that were originally included.

4.2   Inserting Figures

In LaTeX, figures are included using the `graphicx` package. The `\includegraphics` command is used to insert an image. Let's consider an example:

Listing 4.1: A Basic Example of Including a Figure.

```
\usepackage{graphicx}

\begin{figure}[htb]
    \centering
    \includegraphics[width=0.7\linewidth]{example-image}
    \caption{Example Figure}
    \label{fig:example}
\end{figure}
```
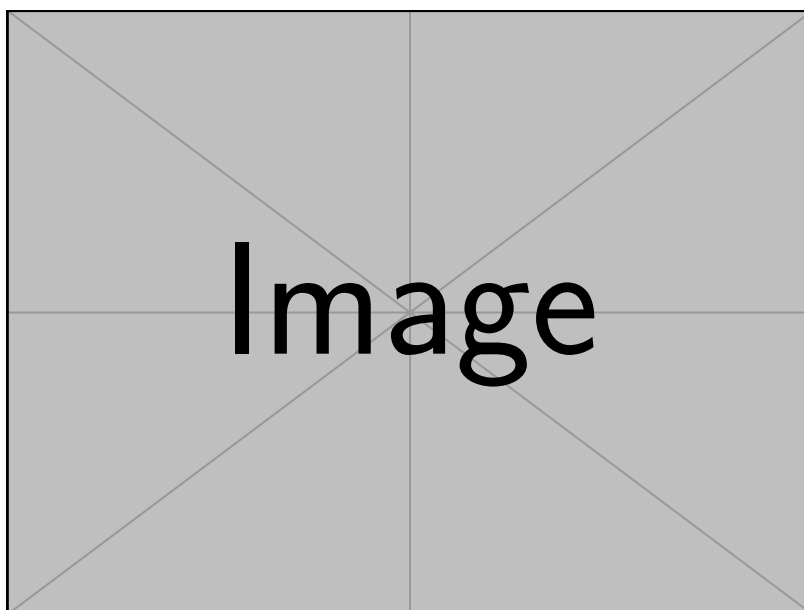

Figure 4.1: This is an example of a single figure similar to that produced by **??**.

In this example, the `figure` environment is used to contain the image. The `\centering` command ensures the image is centred horizontally. The `width` parameter is used to control the size of the image; in this case, it has been set to `0.45\cmd {linewidth}` which will make it fill a space that is 0.45 tiems the width of

the current line. The `\caption` and `\label` commands provide a caption and label for referencing, respectively.

Figures can be formatted to meet specific requirements. The `\subfigure` command from the `subcaption` package can be used for side-by-side figures:

```
\usepackage{subcaption}

\begin{figure}[htb]
    \begin{subfigure}{0.45\linewidth}
        \centering
        \includegraphics[width=\linewidth]{example-image-a}
        \caption{Subfigure A} % Leave blank for just letter
        \label{subfig:a}
    \end{subfigure}
    \hfill
    \begin{subfigure}{0.45\linewidth}
        \centering
        \includegraphics[width=\linewidth]{example-image-b}
        \caption{Subfigure B} % Leave blank for just letter
        \label{subfig:b}
    \end{subfigure}
    \caption{Example with Subfigures}
    \label{fig:subfigures}
\end{figure}
```

This example uses the `subfigure` environment to create subfigures within a larger figure (as show in **??**. The `\hfill` command adds horizontal space between the subfigures.

<div align="center">(a)</div>

<div align="center">(b)</div>
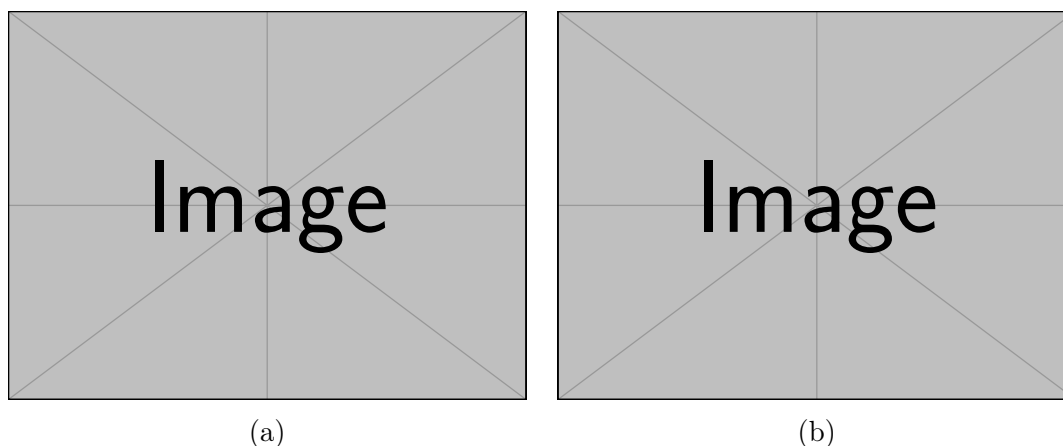
Figure 4.2: This is an example of a double image figure similar to that produced by **??**.

While this section provided a few examples on how to make some figures and sub-figures, I would strongly recommend checking out some of the more complex examples of figures shown in **??**. This will provide the code and examples for how to create more intricate subfigures and layouts.

4.3 Tables and Tabularx

Tables in LaTeX are created using the `tabular` environment. The `tabularx` package is particularly useful when you want the table to automatically adjust its width. Let's define some custom column types for convenience:

```
\usepackage{tabularx}
\newcolumntype{C}{>{\centering\arraybackslash}X}
\newcolumntype{L}{>{\raggedright\arraybackslash}X}
\newcolumntype{R}{>{\raggedleft\arraybackslash}X}
```

Now, let's create a table using tabularx:

In this example, the `tabularx` environment is used, and the custom column types `C`, `L`, and `R` are applied to the columns. This ensures the content is centered, left-justified, and right-justified, respectively.

```latex
\begin{table}[htb]
    \centering
    \begin{tabularx}{\linewidth}{|C|L|R|}
        \hline
        \textbf{Centered} & \textbf{Left Justified} & \textbf{
            Right Justified} \\
        \hline
        Content & More content & Additional content \\
        \hline
    \end{tabularx}
    \caption{Example Table with Tabularx}
    \label{tab:example}
\end{table}
```

4.4   Advanced Table Features

To create professional-looking tables, the `booktabs` package can be employed. It provides commands for better spacing and styling of tables:

```latex
\usepackage{booktabs}

\begin{table}[htb]
    \centering
    \begin{tabular}{ccc}
        \toprule
        \textbf{Header 1} & \textbf{Header 2} & \textbf{Header
            3} \\
        \midrule
        Content 1 & Content 2 & Content 3 \\
        Content 4 & Content 5 & Content 6 \\
        \bottomrule
    \end{tabular}
    \caption{Example Table with Booktabs}
    \label{tab:booktabs_example}
\end{table}
```

The `\toprule`, `\midrule`, and `\bottomrule` commands create horizontal rules with appropriate spacing.

4.5   Additional Packages for Enhanced Table Functionality

Several other packages can be employed to enhance table functionality:

The `longtable` package allows tables to span multiple pages, which is useful for large datasets. The `multirow` and `multicolumn` packages provide commands for cells that span multiple rows or columns, respectively. The `makecell` package enables more complex table layouts. Each of these packages comes with its set of commands and options. Let's briefly explore the usage of `longtable`, `multirow`, and `multicolumn`:

In these examples, the `longtable` environment is used for tables that span multiple pages. The `\multirow` command is employed to create cells that span multiple rows, while `\multicolumn` is used for cells that span multiple columns.

4.6   Conclusion

This Chapter provided a comprehensive overview of including figures and tables in your LATEX document. From basic insertion of figures to advanced table formatting using packages like `tabularx`, `booktabs`, and others, you now have a solid foundation to include tables and figures in your thesis. While this provides a lot of details on how to add figures that have already been pre-generated, one might want to generate figures on the spot potentially even using data generated from other programs (such as MatLab®, Python, *etc.*). For this **??** provides in-depth workings of the `pgfplots` package and how to generate consistent and professional looking plots and graphs.

```latex
\usepackage{longtable}
\usepackage{multirow}
\usepackage{multicolumn}

% Example Longtable
\begin{longtable}{|c|c|}
    \caption{Longtable Example} \label{tab:longtable} \\
    \hline
    \textbf{Header 1} & \textbf{Header 2} \\
    \hline
    \endfirsthead
    \hline
    \textbf{Header 1} & \textbf{Header 2} \\
    \hline
    \endhead
    Content 1 & Content 2 \\
    Content 3 & Content 4 \\
    \hline
\end{longtable}

% Example Multirow and Multicolumn
\begin{table}[htb]
    \centering
    \begin{tabular}{|c|c|c|}
        \hline
        \multirow{2}{*}{\textbf{Multirow-Col1}} & \multicolumn
            {2}{c|}{\textbf{Multicolumn-Col2-3}} \\
        \cline{2-3}
        & \textbf{Column 2} & \textbf{Column 3} \\
        \hline
        Content 1 & Content 2 & Content 3 \\
        \hline
    \end{tabular}
    \caption{Example Table with Multirow and Multicolumn}
    \label{tab:multirow_multicolumn}
\end{table}
```

Chapter 5

Plots, Charts, & Graphs

Throughout this chapter we will be exploring some of the different ways of displaying your data in your thesis. Mainly this will be accomplished with the `pgfplots` package. In the following sections, there will be a few examples of how to generate different plots. For more information on how to create plots, here is the manual for `pgfplots`(the package used to generate the information for TikZ to create the plots). For an extensive list of examples please refer to this

5.1 Line Plots

A simple line plot can be effectively created using the `axis` environment from the `pgfplots` package in L^AT_EX. The `pgfplots` package is a powerful tool for creating high-quality plots directly within L^AT_EX documents. It builds upon the `TikZ` package and provides a comprehensive set of options for plotting and customizing graphs.

The following code (see **??**) can be used to create the figure shown in **??**. Expanding on this example, we can add a second plot by adding the following code below the closing bracket and semi-colon (`};`) of the `\addplot` command: This will result in the addition of the second red line shown in **??**.

```
\begin{figure}[htbp]
    \centering
    \begin{tikzpicture}
        \begin{axis}[
            title={Simple Line Plot},
            xlabel={X-axis},
            ylabel={Y-axis},
        ]
            \addplot coordinates {
                (0,0)
                (1,1)
                (2,4)
                (3,9)
                (4,16)
            };
        \end{axis}
    \end{tikzpicture}
    \caption{A simple line plot.}
    \label{fig:line-plot}
\end{figure}
```

(a)
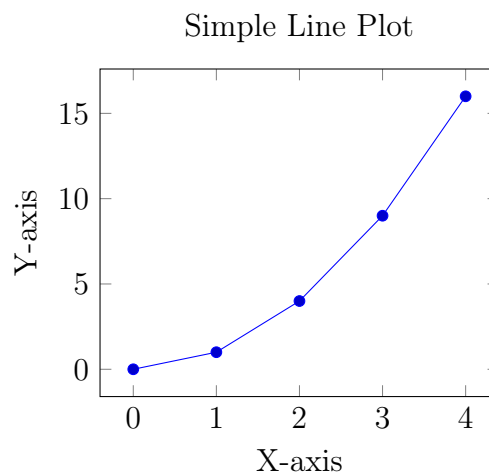
(b)

Figure 5.1: A simple line plot (??) and the code to generate the plot (??)

```
\begin{tikzpicture}
    \begin{axis}[
        title={Dual Line Plot},
        xlabel={X-axis},
        ylabel={Y-axis},
    ]
        \addplot coordinates {
            (0,0)
            (1,1)
            (2,4)
            (3,9)
            (4,16)
        };
        \addplot coordinates {
            (0,16)
            (1,9)
            (2,4)
            (3,1)
            (4,0)
        };
    \end{axis}
\end{tikzpicture}
```

(a)
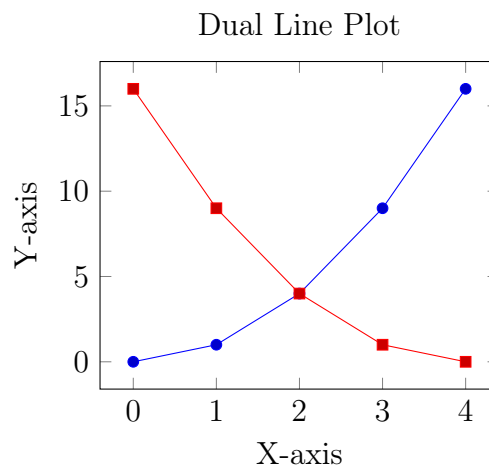
(b)

Figure 5.2: A simple line plot with two sets of data.

```
\addplot coordinates {
    (0,16)
    (1,9)
    (2,4)
    (3,1)
    (4,0)
};
```

5.2   Customizing Plots

This section provides some ways to increase the readability and customization of the plots we generate.

5.2.1   Adding a Legend

Legends can be added to plots for better readability. To add a legend to your plot you can use the code in **????** to generate the plot shown in **??**. Each plot is individually added to the legend by adding a `\addlegendentry`{YOUR LEGEND ENTRY HERE} command following the `\addplot` command.

Note: *The position of the legend can be specified by using the optional parameter* `legend pos=` followed by a set of compass coordinates.

```
\begin{figure}[H]
    \centering
    \begin{tikzpicture}
        \begin{axis}[
            title={Plot with Added Legend},
            xlabel={X-axis},
            ylabel={Y-axis},
            legend pos=north west,
        ]
```

```
        \addplot coordinates {
            (0,0)
            (1,1)
            (2,4)
            (3,9)
            (4,16)
        };
        \addlegendentry{\(y = x^2\)}
        \addplot coordinates {
            (0,16)
            (1,9)
            (2,4)
            (3,1)
            (4,0)
        };
        \addlegendentry{\(y = 16 - x^2\)}
    \end{axis}
  \end{tikzpicture}
  \caption{A customized plot with a legend.}
  \label{fig:legend-plot}
\end{figure}
```

Plot with Added Legend
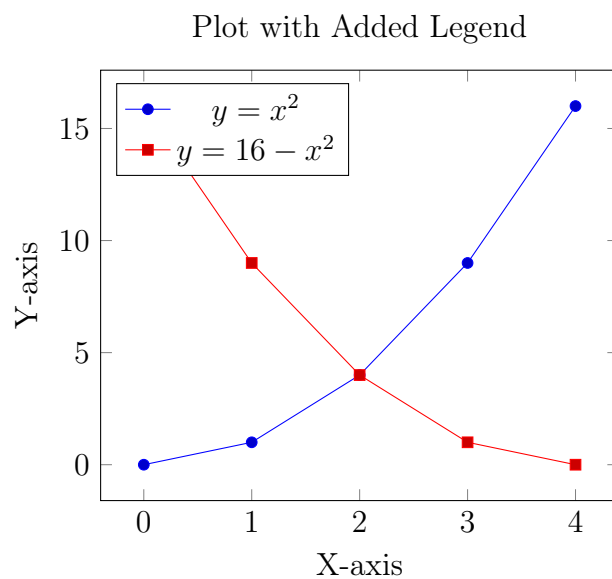
Figure 5.3: A customized plot with a legend.

5.2.2 Adding Grid Lines

To add gridlines to your plot
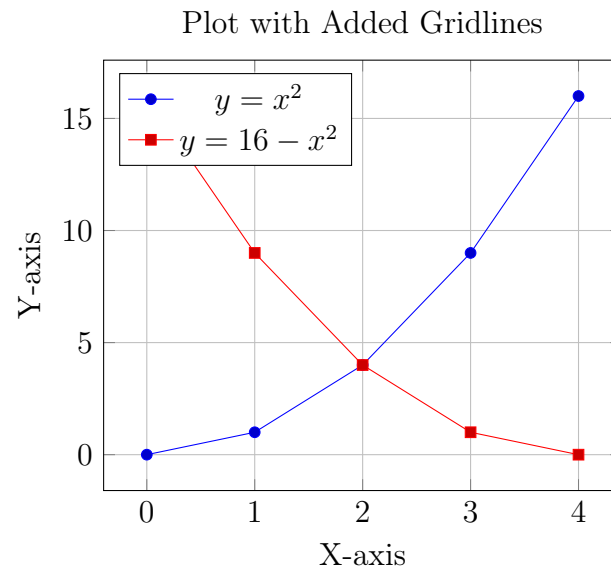
Figure 5.4: A customized plot with added gridlines.

5.2.3 Changing Colors and Line Styles

Colors and line styles can be easily modified:

Colored and Styled Plot
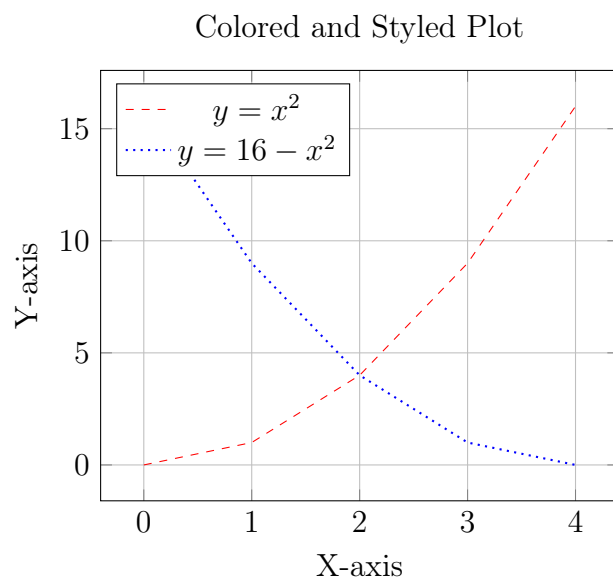
Figure 5.5: A plot with customized colors and line styles

5.3   Advanced Plot Types

5.3.1   Equations
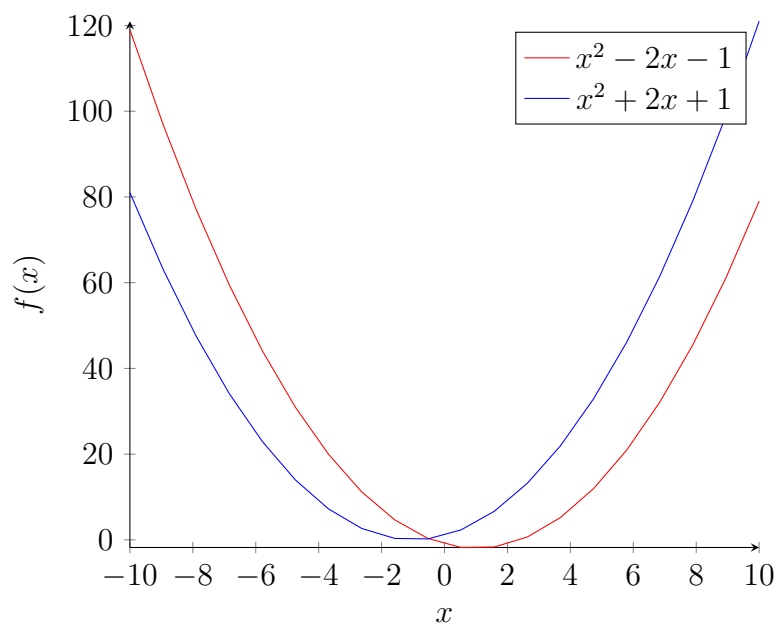
Figure 5.6: Plot of two parabola.

5.3.2 Scatter Plot with External Data
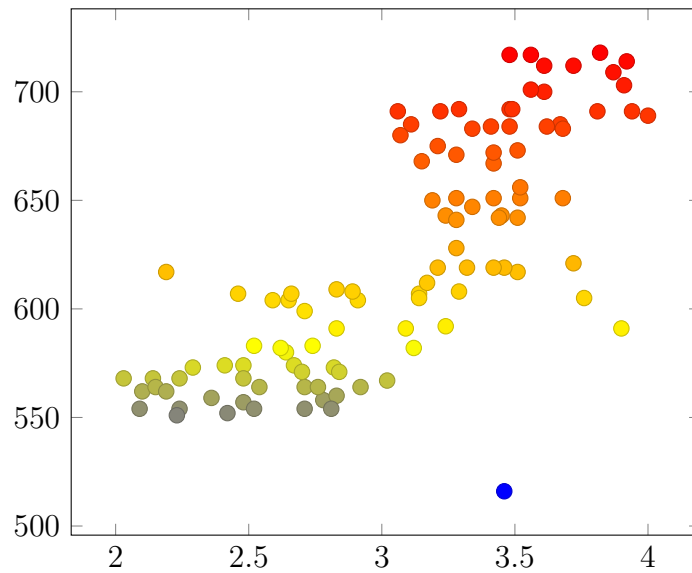
Figure 5.7: Example of a Scatter Plot.

5.3.3 Bar Plot

Bar plots are useful for categorical data. Here's how to create one:
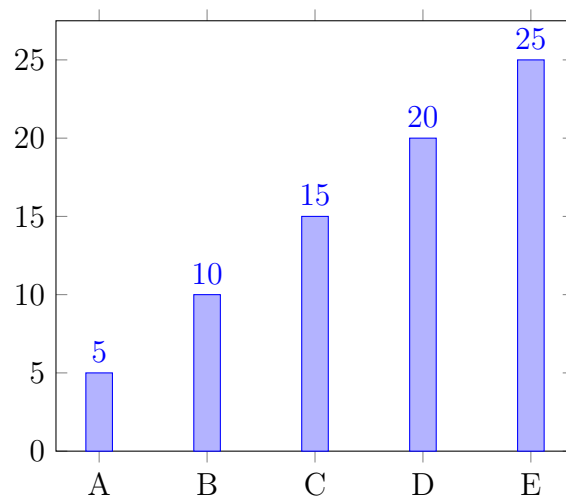
Figure 5.8: A bar plot
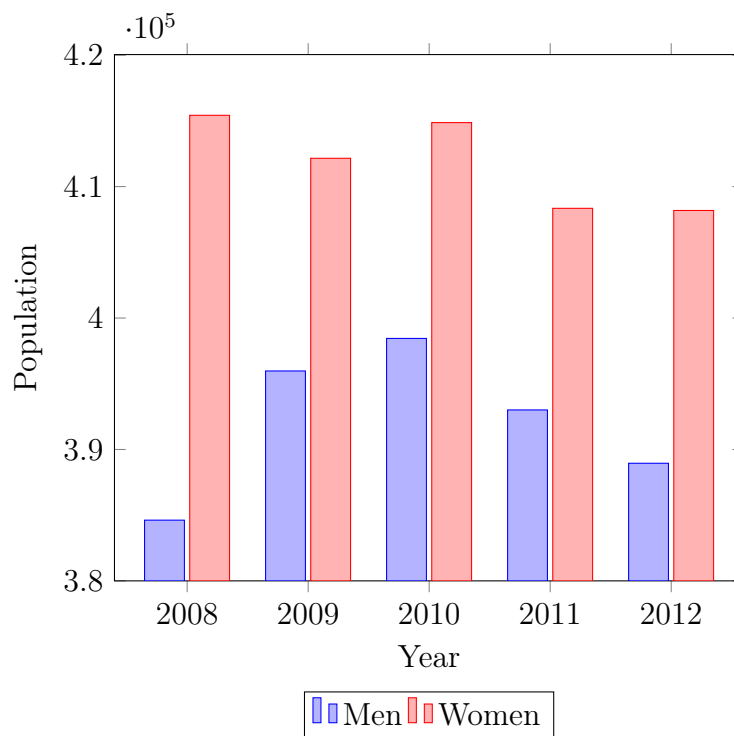
Figure 5.9: Example of a Bar Graph.

5.3.4 Pie Chart

Pie charts are less common in LaTeX, as well as, a lot of other media due to the pour representation of the data. However, if you are still inclined to use them the remainder of this section will show how the package `pgf-pie` can be used to create consistent graphs that look good.
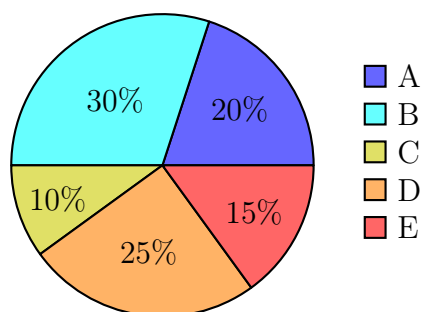
Figure 5.10: A basic pie chart.
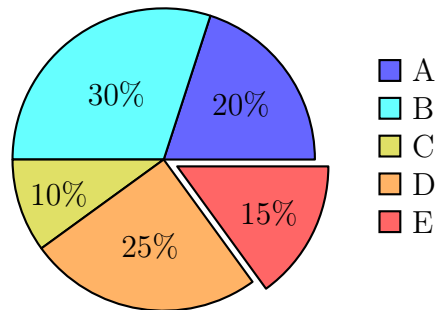
Figure 5.11: A pie chart with an "Exploded" slice.
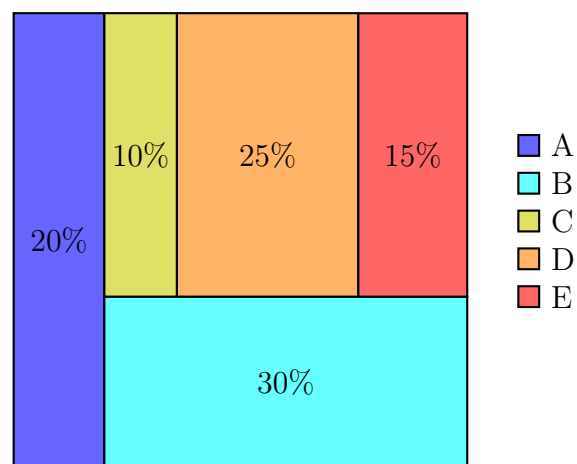
Figure 5.12: A "square" pie chart.

5.3.5   3D Plot

3D plots can be created for more complex data visualization:
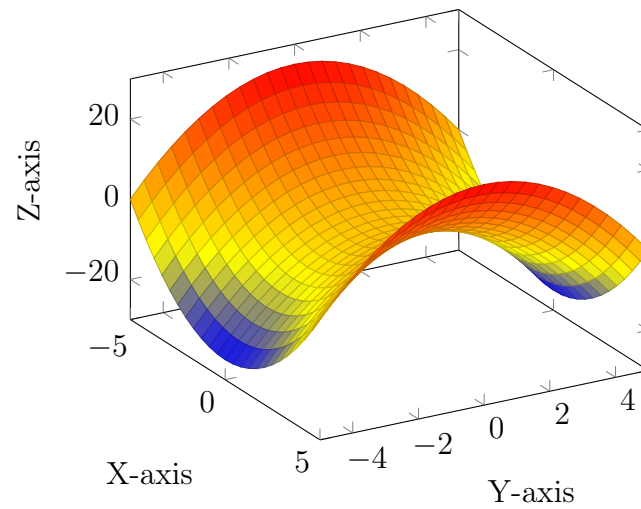
Figure 5.13: A 3D surface plot

Example using the mesh parameter
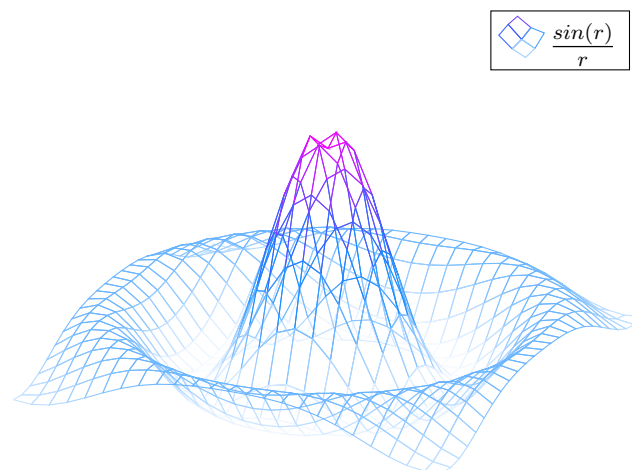
Figure 5.14: Example of a 3D Plot

5.3.6 Polar Plot

Polar plots are useful for circular data:
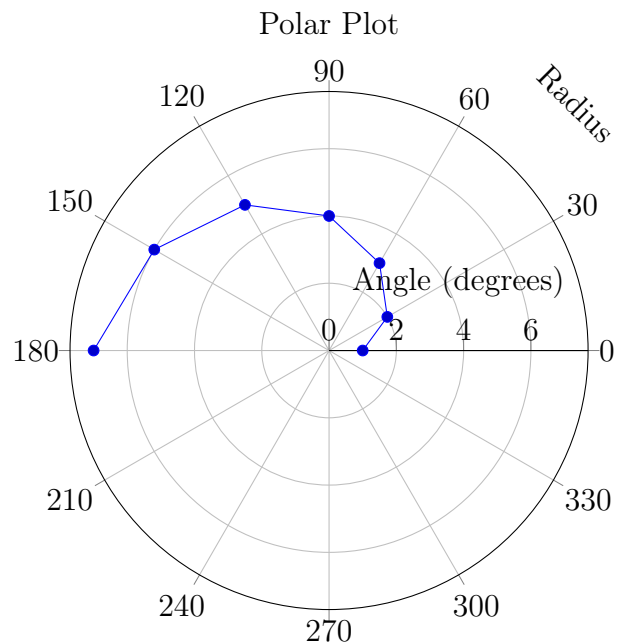
Figure 5.15: A polar plot

5.3.7   Box Plot

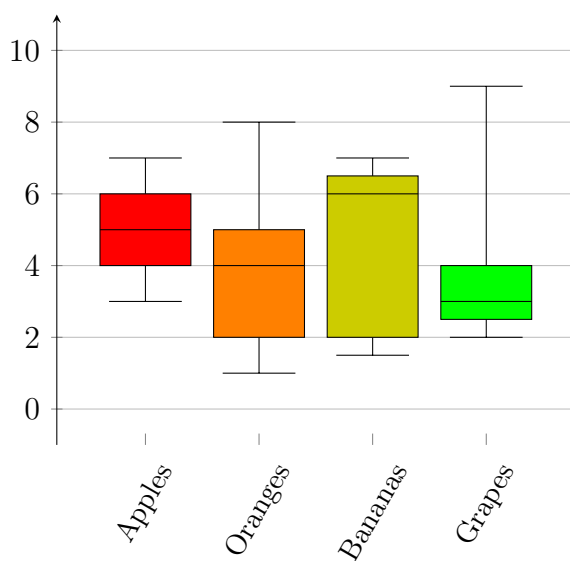Box plots are used to visualize the distribution of data:

Figure 5.16: A box plot

5.4 Conclusion

The `pgfplots` package is an incredibly versatile tool for creating a wide range of plots and graphs in LaTeX. This chapter has provided examples of various plot types and customization options, showcasing the power and flexibility of `pgfplots`. By leveraging these capabilities, you can create high-quality, publication-ready figures for your thesis.

Chapter 6

Mathematical Equations

There are many ways to include formulas in your thesis. This section will provide some different ways of adding them (inline and standalone), as well as provide some ways of referencing the equations.

To start the simplest way to add an equation is using the built-in LaTeX math mode. To enter and exit math mode one just needs to use the `\(` and `\)` symbols around an equation. While there also exists `$<Equation>$` to add math, it is not recommended due to potential compatibility issues. Additionally, this, `\(<Equation>\)`, method is capable of being redefined to add further customization. An example of using math mode to get an inline equation is by using the following command:

$$\(\vec{F_{d}}=\frac{1}{2}\ A\ C_{d}\ \vec{V}^{2}\)$$

The above command has the effect of creating the following output: $\vec{F_d} = \frac{1}{2}\ A\ C_d\ \vec{V}^2$. Sometimes it can be quite beneficial to separate what would be an inline equation to be on its own line. For this, we have two different ways of doing it. The first was will produce an equation that has no reference:

$$E = m\ c^2$$

$$E = m\ c^2$$

The second will produce an equation with a reference. For this, there are two main ways of creating the reference, the first one, see **??**, creates a numbered reference; the other one, see **??**, creates a reference with a 'tag'. The difference between the two is the inclusion of a `\tag{<text>}` command that will replace the regular number with `<text>` and the changing from the `equation` environment to the `equation*` environment. If you do not want the brackets around the tag, as shown in **??**, use the starred version of the command: `\tag*{<text>}`. This will not remove the braces in the reference for the equation, but will remove them from appearing next to the equation definition.

NOTICE Using the `\tag` command in conjunction with the `\label` command can create a LaTeX warning when used in the non-star `equation` environment. This warning can be safely ignored, however, the better way to deal with this is to make sure one is using the star version, `equation*`.

$$\pi = 3.14... \tag{6.1}$$

$$\pi = 3.1415... \tag{Constant pi}$$

$$\pi = 3.1415... \tag*{Constant pi}$$

If you have multiple equations that you want arranged very neatly, use the `align` environment and you can assign individual equations numbers as shown in **??????**. Note that it is the & symbol that determines what will be aligned. Further note that spaces in "math mode" are ignored and need to be specified using the space commands in

$$Equation1 = 1 \tag{6.2}$$

$$Equation2 = 2 + 2 \tag{6.3}$$

$$Equation3 = 3 + 3 + 3 \tag{6.4}$$

It may be very important in a math heavy thesis to be able to show your equations,

or even data in a readable way. For this, we will explore some of the ways to create specific data.

6.1 Vector, Sets, Piecewise Functions, Matrix Math, and More

$$
f(x) = \begin{cases} x^{2*\ln x}, & \text{if } x < 3 \\ -\frac{x}{2}, & \text{if } 3 \le x \le 4 \\ x, & \text{if } 4 < x \end{cases} \tag{6.5}
$$

Vectors and Matrices are used in many fields of math and science and provide a convenient way to represent 2-Dimensional arrays of numbers.

$$
x \in \{1, 2, 3, 4, 5, 6, 7\} \tag{6.6}
$$

$$
V_1 = \begin{pmatrix} a, & b, & c, & d \end{pmatrix} \tag{6.7}
$$

$$
V_2 = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \tag{6.8}
$$

$$
M = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \tag{6.9}
$$

$$
p(x) = 3x^6 + 14x^5 y + 590x^4 y^2 + 19x^3 y^3
$$

$$
- 12x^2 y^4 - 12xy^5 + 2y^6 - a^3 b^3
$$

$$
ralign = lalign \tag{6.10}
$$

$$
ralign = lalign \tag{6.11}
$$

$$x = y \qquad\qquad w = z \qquad\qquad a = b + c$$

$$2x = -y \qquad\qquad 3w = \frac{1}{2}z \qquad\qquad a = b$$

$$-4 + 5x = 2 + y \qquad\qquad w + 2 = -1 + w \qquad\qquad ab = cb$$

m	Modulo	$m > 0$
a	Multiplier	$0 < a < m$
c	Constant	$0 \leq c < m$
x_0	Initial Value	$0 \leq x_0 < m$

m	Modulo	$m > 0$
a	Multiplier	$0 < a < m$
c	Constant	$0 \leq c < m$
x_0	Initial Value	$0 \leq x_0 < m$

$$eqn1 = fdsjalk \tag{6.12}$$

$$eqn2 = fdsa * 243 \tag{6.13}$$

$$eqn$$

$$= eqn \tag{6.14}$$

6.2   Functions

`\sin`	sin	`\cos`	cos	`\tan`	tan
`\arcsin`	arcsin	`\arccos`	arccos	`\arctan`	arctan
`\sec`	cos	`\csc`	csc	`\cot`	cot
`\sinh`	sinh	`\cosh`	cosh	`\tanh`	tanh
`\ln`	ln	`\log`	log	`\exp`	exp

6.3

6.4   Vector, Sets, Piecewise Functions, Matrix Math, and More

Chapter 7

Citations, References, and Cross-References

This section will be showing off some of the different ways to include "citations" and "cross-references" within your document. Note that **cross-references** in LaTeX utilize `\ref` as a command, while one might think that this is short for reference this is not the case citation/references utilize the `\cite` command.

7.1   Cross-References

In LaTeX, references will "reference" a `\label{Reference:Label}` command. This section has the following command to define the Chapter:

` \chapter{Citations/References, and Cross-References}\label{ch:citref}`

By using `\ref{ch:citref}`|, this allows you to insert a cross-reference that look like this **??**. Now this by itself is not the most useful, to make it a bit better we should keep track of what we are cross-referencing, in this case a **Chapter**, and add this label in front of the cross-reference (`Chapter~\ref{sec:citref}`) and this will display like this: Chapter **??**.

Note: *To ensure the reference is not split we need add a non-breaking space ($\sim$) to prevent LaTeX from accidentally adding a line-break between the label we added and the actual cross-reference.*

While using the `\ref` command, you might ask "*Why does LaTeX not just know what it is that I am referencing and insert that automatically in front of the reference?*" The answer is to provide more flexibility to the user. However, that being said, individuals have created a number of packages that work to enhance the workflow of adding these cross-references. Some of these are provided by the `hyperref` and `cleveref` packages. To include these packages add the following lines to the **bottom** of your preamble (order matters, `cleveref` needs to be after `hyperref`; and `hyperref` should be one of the last packages loaded):

```
\usepackage{hyperref}
\usepackage[nameinlink]{cleveref}
```

With these packages loaded we can now use the commands listed in **??**.[1]

---

[1]Note that because the floats are added where they are in the text this causes them to insert large amounts of white space because it only fits on the following page.

Table 7.1: Built-in, hyperref, and cleveref commands and outputs

Command	Output
built-in	
\ref{}	??
\pageref{}	??
hyperref	
\autoref{}	??
cleveref	
\cref{}	??
\Cref{}	??
\cref*{}	??
\Cref*{}	??
\cpageref{}	??
\Cpageref{}	??
\namecref{}	??
\nameCref{}	??

Further, the `cleveref` also includes features that allows for the auto sorting and combining of references:

```
\Cref{fig:doubleImage,fig:singleImage,fig:
    tripleImage1,fig:quadImage}
```

Noting that there are **NO** spaces between the labels; this will produce: **????????**. Allowing one to quickly and efficiently keep references up-to-date and consistent in their style. More examples of the use of the `cleveref` cross-referencing is found through the rest of this **??**.

7.2 Citations/References

7.3 Citation Managers

7.3.1 JabRef

Information on the use of this has been moved to it's own Chapter

7.4 This is old Material

This section will be showing off some of the different ways to include "citations" and "cross-references" within your document. Note that **cross-references** in LaTeX utilize `\ref` as a command, while one might think that this is short for reference this is not the case citation/references utilize the `\cite{}` commands.

7.4.1 Cross-References

In LaTeX, references will "reference" a `\label{Reference:Label}` command. This section has the following command to define the the section:

> `\subsection{Cross-References}\label{subsec:cross-reference}`

By using `\ref{subsec:cross-reference}`, this allows you to insert a reference that look like this: **??**. Now this by itself is not the most useful, to make it a bit better we should keep track of what we are referencing, in this case a **Section**, and add this label in front of the reference (`Section~\ref{subsec:cross-reference}`) and this will display like this: Section **??**. Note to ensure the reference is not split we add a non-breaking space (`~`) to prevent LaTeX from adding a line break.

While using the ref command, you might ask " *Why does LaTeX not just know what it is that I am referencing and insert that automatically in front of the reference?*" The answer is to provide more flexibility to the user. However, that being said,

individuals have created a number of packages that work to enhance the workflow of adding these cross-references. Some of these are provided by the `hyperref` and `cleveref` packages. To include these packages add the following lines to the bottom of your preamble (order matters, cleveref needs to be after hyperref and hyperref should be one of the last packages loaded):

```
\usepackage{hyperref}
\usepackage[nameinlink]{cleveref}
```

With these packages installed we can now use the commands in **??**.[2]

Table 7.2: Built-in, hyperref, and cleveref commands and outputs

Command	Output
built-in	
\ref{}	??
\pageref{}	??
hyperref	
\autoref{}	??
cleveref	
\cref{}	??
\Cref{}	??
\cref*{}	??
\Cref*{}	??
\cpageref{}	??
\Cpageref{}	??
\namecref{}	??
\nameCref{}	??

---

[2]Note that because the floats are added where they are in the text this causes them to insert large amounts of white space because it only fits on the following page.

Further, the `cleveref` also includes features that allows for the auto sorting and combining of references:

```
\Cref{fig:doubleImage,fig:singleImage,fig:
        tripleImage1,fig:quadImage}
```

Noting that there are **NO** spaces between the labels; this will produce: **????????**. Allowing one to quickly and efficiently keep references up-to-date and consistent in their style. More examples of the use of the `cleveref` cross-referencing is found through the rest of this document.

7.4.2 Citations

Citations are a lot easier than dealing with the cross-referencing. There are no additional packages required for citations, the built-in ones are feature-rich enough. Now, while there are no additional packages required to make citations in your document, there are in fact a few programs that should help you manage all of your citations/references. These programs can include Mendeley, JabRef, or Zotero; a comparison of the softwares can be found in **??**, and more information of the use of JabRef can be found in **??**. Single citations can be included with the `\cite{citationKey}` command, the

Table 7.3: Comparison of Reference Softwares

Software	Developer	Version	Cost	License
JabRef	The JabRef Team	5.11	Free	MIT
Mendeley	Elsevier	2.99.0	Free up to 2 GB	Proprietary
Zotero	CDS	6.0.27	Free up to 300 MB	AGPL

one at the end of this sentence is created with the `\cite{TEST}` command[**TEST**]. Multiple citations can be included in a single cite command by adding commas in between the citation keys. The citation at the end of this sentence shows how to

create more than one citation and how they are grouped together, it is created with the `\cite{testone,cite2,cite3,cite4,cite5}` command[**testone**, **cite2**, **cite3**, **cite4**, **cite5**]. Finally this sentence shows how a gap in the citations is handled, this is created with the `\cite{testone,cite2,cite3,cite5}` command[**testone**, **cite2**, **cite3**, **cite5**].

Chapter 8

Submitting Your Thesis

So you have seemingly gotten to the end of the writing, and you may be already taking the steps to set up your last review with your supervisor, set up your thesis defence, or even submit your thesis to GPS... but now what do you do?

Quick answer is a lot, long answer will be discussed throughout this Chapter.

There are a number of steps that you will want to take to make sure that you are submitting the best version of your work. This includes checking for some of the more obvious and less obvious pitfalls that writing a Thesis in LaTeX or really any software poses.

Chapter 9

JabRef: Managing Bibliographies Efficiently

9.1 Introduction

JabRef stands as a powerful tool for researchers and academics engaged in scholarly writing. JabRef offers a robust solution for bibliography management, including a number of features to ensure that you are not only able to organize your references but keep track of progress, and notes on each reference. This chapter aims to provide an review of JabRef, including its myriad features that I find particularly useful, and to guide you through its implementation in writing your thesis.

9.2 Key Features of JabRef

JabRef, with its versatile features, emerges as an indispensable tool for bibliography/reference management. Delving deeper into its functionalities reveals a wealth of tools designed to streamline the often cumbersome process of handling references.

9.2.1 BibTeX Compatibility

JabRef's commitment to the BibTeX format is a testament to its roots in the LaTeX ecosystem. This compatibility ensures a seamless integration between the reference

management process and the LaTeX document preparation workflow. Users can easily export and import BibTeX files, facilitating collaboration and compatibility across various platforms.

9.2.2 Reference Import

The capability to import references directly from online databases and journal websites significantly accelerates the reference collection process. JabRef supports various import formats, allowing users to effortlessly populate their databases with accurate and structured reference information. This feature is particularly valuable for researchers dealing with large/extensive bibliographies.

9.2.3 Customizable Entry Types

The flexibility offered by customizable entry types allows users to categorize references based on the nature of the source. Whether it's a book, article, conference proceeding, or any other reference type, JabRef accommodates diverse sources, ensuring a well-organized and easily navigable bibliography.

9.2.4 Search and Filter

The ability to efficiently search and filter references is a hallmark of JabRef's usability. Researchers dealing with extensive databases will appreciate the quick and precise retrieval of references based on author names, titles, keywords, or any other criteria. This feature is crucial for maintaining order in a rapidly growing bibliography.

9.2.5 Grouping

JabRef's grouping functionality provides a systematic approach to organizing references. Users can create custom groups to categorize references based on themes, projects, or any other criteria. This feature is especially useful for large research

projects where a systematic organization of references is essential for maintaining clarity and coherence.

9.2.6   Integration with LaTeX

The seamless integration of JabRef with LaTeX editors fortifies the synergy between bibliography management and document preparation. This integration minimizes the manual effort required for citation insertion and ensures consistency between the bibliography and the in-text citations. Users can easily copy citation keys from JabRef and paste them directly into their LaTeX documents.

9.3   Getting Started with JabRef

Now that we've outlined the key features of JabRef, let's embark on a comprehensive guide on how to get started with JabRef. This step-by-step walkthrough will cover everything from installation to creating a new bibliography and populating it with references.

9.3.1   Installation

The initial step in utilizing JabRef is to install the software on your system. For all users, regardless of OS, the easiest way to download JabRef is to visit their website: https://www.jabref.org/. Once there select 'Download' from the navigation bar, and press the "Download JabRef" button. This will take you to the FossHub page where you can select the appropriate version for your OS and download and install it.

9.3.2   Creating a New Bibliography

Once JabRef is successfully installed, launch the application. When the program loads you will be faced with a window that looks like the one shown in **??**. Now that the program is open, to create a new bibliography:
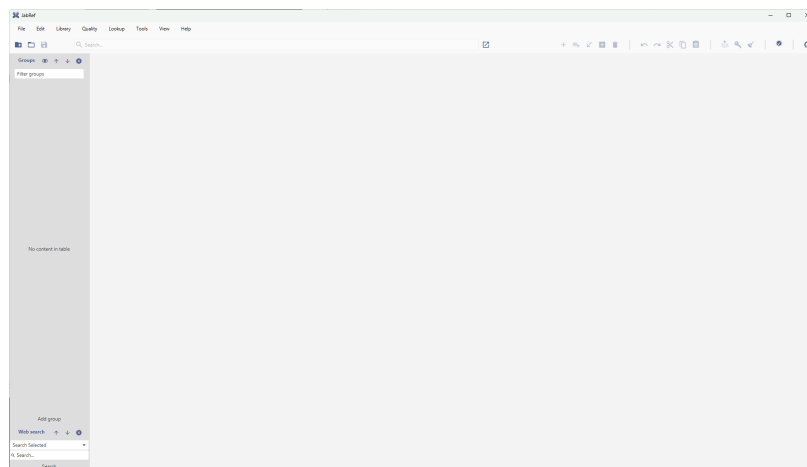
Figure 9.1: JabRef Main Window.

1. Click on 'File → New Library'.

To save the database:

1. Click on 'File → Save Library'.

2. Choose an appropriate and location.

3. Click 'Save'.

Congratulations! You've initiated your bibliography using JabRef. Now that we have this created, the next step is to add references to the database.

9.3.3   Adding References

JabRef offers multiple avenues for adding references to your database. Some of the methods are generally more useful than others but we will go over a few that you are likely to use:

9.3.3.1   Web Search

JabRef's integrated web search (see **??**) feature simplifies the process of importing references from online sources. This is by-far the easiest way to enter a reference.
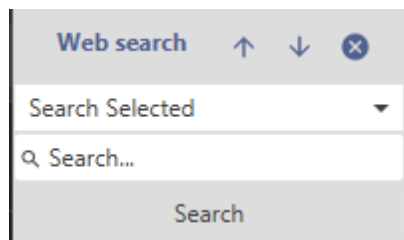
Figure 9.2: JabRef Web Search Tool.

1. Click on 'Web Search'.

2. Search for the desired reference using the integrated search feature.

3. Select the reference all the references you wish to import,as shown in **??**.

4. Click 'Import entries' to import the selected entries.

5. The references are added to your library.
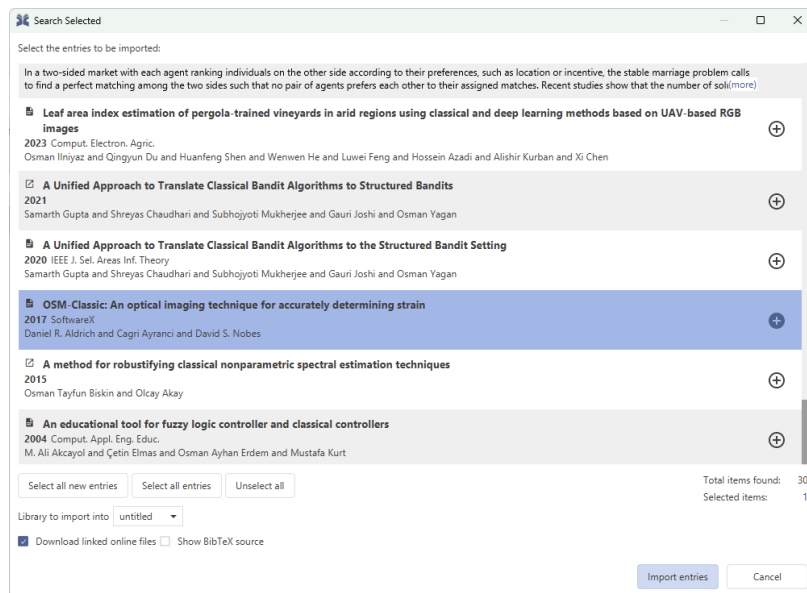
Figure 9.3: Example Web Search Results for ''`OSM-Classic`''.

Note: *The Web Search tool by default uses a general search, however, a specific database can be chosen as well using the drop down arrow next to "Search Selected".*

9.3.3.2 Manual Entry

Manually adding a reference can be done in a 'manual' and 'automatic' way. When adding a new entry you will be faced with the option to select an entry type or to enter an ID (DOI, ArXiv, ISBN, *etc.*). If you enter an ID, the information for the reference will be automatically pulled from the internet. Otherwise to manually enter all the information for a reference:

1. Click Library → 'New entry' or use the shortcut 'Ctrl + N' and this will show the following window.
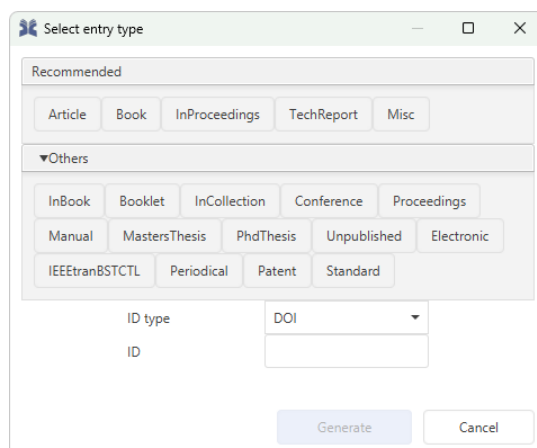
2. Choose the entry type (*e.g.*, article, book, inproceedings).

3. Fill in the required fields like author, title, journal, *etc.*

By following these steps, you can efficiently populate your JabRef database with the necessary references.

9.3.4 Organizing References

Effectively organizing references is essential for a streamlined bibliography. JabRef's grouping feature allows you to categorize references based on your preferences:

1. On the left panel, select 'Add Group'.

2. Give the group an appropriate name.

3. Optionally you can add a Description, Icon, Colour, *etc.*

To assign a reference to a group:

1. Select the Reference(s) from the centre list.

2. Drag them to the group on the left of the screen.

Note: *Groups can even be nested into groups to provide more levels of organization.* Organizing references into groups enhances accessibility and facilitates a more systematic approach to bibliography management.

9.4 Exploring Advanced Features of JabRef

JabRef's capabilities extend beyond the basics covered in the previous sections. In this section, we'll explore some of the advanced features that enhance the efficiency and effectiveness of JabRef as a reference manager.

9.4.1 Quality Assurance: Checking and Correcting Entries

Ensuring the accuracy and completeness of references is crucial. JabRef provides tools for quality assurance, allowing users to check and correct entries.

To check for duplicate entries:

1. Click on 'Quality → Find duplicates'.

2. JabRef will identify and display duplicate entries.

To correct entries:

1. Click on 'Quality → Cleanup entries'.

2. JabRef will provide some useful option to ensure conformity within the different references. This includes renaming Linked PDF's to match the standard of ``CitationKey - Title''.

These quality assurance features contribute to maintaining a clean and error-free bibliography.

9.4.2   Managing PDFs and File Links

JabRef facilitates the management of associated PDF's and file links, offering a consolidated approach to reference and document management.

To link a PDF or file:

1. Open the entry editor for a reference.

2. Click on 'General' and use the 'PDF' or 'File' field to link the document.

This integration helps to streamline the retrieval of PDF's or other associated documents directly from JabRef. Further, this allows JabRef to keep track of the comments and highlights in a single place (see **??**). These annotations can be found by selecting the entry, and selecting the 'File annotations' tab.
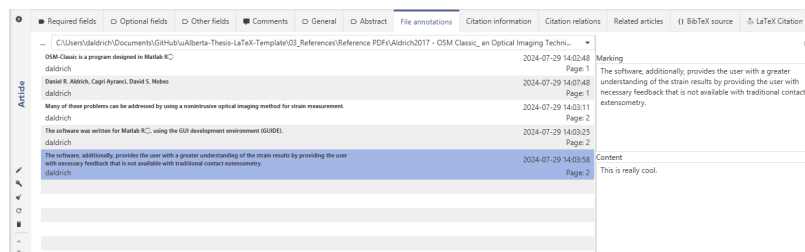
Figure 9.4: Showcase of the file annotations in JabRef.

9.4.3   Additional Information

JabRef keeps track of a lot of information and can even help you with your research. Some additional information JabRef provides includes:

- Citation information

- Citation relationships (what the reference cites and who has cited the reference). This further lets you open the links to the reference's source, or even add these references directly to your library.

- If one right clicks an entry you are provided the following options:

  - Rank - rank the reference with one to five stars.

  - Toggle Relevance - add a marker to show this is a relevant source.

  - Priority - rank items as low, medium, or high priority.

  - Read Status - set the status to read or skimmed.

Bibliography

[0]   W. H. Ray, *Advanced process control*. McGraw-Hill: New York, NY, USA, 1981.

[0]   E. Davison, "The robust control of a servomechanism problem for linear time-invariant multivariable systems," *IEEE transactions on Automatic Control*, vol. 21, no. 1, pp. 25–34, 1976, ISSN: 0018-9286. DOI: 10.1109/tac.1976.1101137.

[0]   A. A. Moghadam, I. Aksikas, S. Dubljevic, and J. F. Forbes, "Infinite-dimensional LQ optimal control of a dimethyl ether (DME) catalytic distillation column," *Journal of Process Control*, vol. 22, no. 9, pp. 1655–1669, 2012, ISSN: 0959-1524. DOI: 10.1016/j.jprocont.2012.06.018.

[0]   D. Dochain, "State observers for tubular reactors with unknown kinetics," *Journal of process control*, vol. 10, no. 2-3, pp. 259–268, 2000, ISSN: 0959-1524. DOI: 10.1016/s0959-1524(99)00020-7.

[0]   D. Dochain, "State observation and adaptive linearizing control for distributed parameter (bio) chemical reactors," *International Journal of Adaptive Control and Signal Processing*, vol. 15, no. 6, pp. 633–653, 2001, ISSN: 0890-6327. DOI: 10.1002/acs.691.

[0]   A. A. Alonso, I. G. Kevrekidis, J. R. Banga, and C. E. Frouzakis, "Optimal sensor location and reduced order observer design for distributed process systems," *Computers & chemical engineering*, vol. 28, no. 1-2, pp. 27–35, 2004, ISSN: 0098-1354. DOI: 10.1016/s0098-1354(03)00175-3.

[0]   J. M. Ali, N. H. Hoang, M. A. Hussain, and D. Dochain, "Review and classification of recent observers applied in chemical process systems," *Computers & Chemical Engineering*, vol. 76, pp. 27–41, 2015, ISSN: 0098-1354. DOI: 10.1016/j.compchemeng.2015.01.019.

[0]   P. D. Christofides and P. Daoutidis, "Feedback control of hyperbolic PDE systems," *AIChE Journal*, vol. 42, no. 11, pp. 3063–3086, 1996, ISSN: 0001-1541. DOI: 10.1002/aic.690421108.

[0]   M. Krstic and A. Smyshlyaev, "Backstepping boundary control for first-order hyperbolic PDEs and application to systems with actuator and sensor delays," *Systems & Control Letters*, vol. 57, no. 9, pp. 750–758, 2008, ISSN: 0167-6911. DOI: 10.1016/j.sysconle.2008.02.005.

[0] X. Xu and S. Dubljevic, "The state feedback servo-regulator for countercurrent heat-exchanger system modelled by system of hyperbolic PDEs," *European Journal of Control*, vol. 29, pp. 51–61, 2016, ISSN: 0947-3580. DOI: 10.1016/j.ejcon.2016.02.002.

[0] P. D. Christofides, "Robust control of parabolic PDE systems," *Chemical Engineering Science*, vol. 53, no. 16, pp. 2949–2965, 1998, ISSN: 2324-9749. DOI: 10.1007/978-1-4612-0185-4_5.

[0] S. Dubljevic and P. D. Christofides, "Predictive control of parabolic PDEs with boundary control actuation," *Chemical Engineering Science*, vol. 61, no. 18, pp. 6239–6248, 2006, ISSN: 0009-2509. DOI: 10.1016/j.ces.2006.05.041.

[0] S. Khatibi, G. O. Cassol, and S. Dubljevic, "Model predictive control of a non-isothermal axial dispersion tubular reactor with recycle," *Computers & Chemical Engineering*, vol. 145, p. 107 159, 2021, ISSN: 0098-1354. DOI: 10.1016/j.compchemeng.2020.107159.

[0] R. Curtain and H. Zwart. Springer Nature, 2020. DOI: 10.1007/978-1-0716-0590-5_1.

[0] M. Krstić (Systems & control). Birkhäuser, 2009.

[0] S. Hiratsuka and A. Ichikawa, "Optimal control of systems with transportation lags," *IEEE Transactions on Automatic Control*, vol. 14, no. 3, pp. 237–247, 1969, ISSN: 0018-9286. DOI: 10.1109/TAC.1969.1099175.

[0] L. Mohammadi, I. Aksikas, S. Dubljevic, and J. F. Forbes, "LQ-boundary control of a diffusion-convection-reaction system," *International Journal of Control*, vol. 85, no. 2, pp. 171–181, 2012, ISSN: 0959-1524. DOI: 10.1080/00207179.2011.642308.

[0] G. O. Cassol and S. Dubljevic, "Discrete output regulator design for a mono-tubular reactor with recycle," in *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 1262–1267. DOI: 10.23919/acc.2019.8815250.

[0] G. O. Cassol, D. Ni, and S. Dubljevic, "Heat exchanger system boundary regulation," *AIChE Journal*, vol. 65, no. 8, e16623, 2019, ISSN: 0001-1541. DOI: 10.1002/aic.16623.

[0] J. Qi, S. Dubljevic, and W. Kong, "Output feedback compensation to state and measurement delays for a first-order hyperbolic PIDE with recycle," *Automatica*, vol. 128, p. 109 565, 2021, ISSN: 0005-1098. DOI: 10.1016/j.automatica.2021.109565.

[0] O. Levenspiel, *Chemical reaction engineering*. John Wiley & Sons, 1998.

[0] K. F. Jensen and W. H. Ray, "The bifurcation behavior of tubular reactors," *Chemical Engineering Science*, vol. 37, no. 2, pp. 199–222, 1982, ISSN: 0009-2509. DOI: 10.1016/0009-2509(82)80155-3.

[0] P. V. Danckwerts, "Continuous flow systems: Distribution of residence times," *Chemical Engineering Science*, vol. 2, no. 1, pp. 1–13, 1953, ISSN: 0009-2509. DOI: 10.1016/0009-2509(53)80001-1.

[0] B. Moadeli, G. O. Cassol, and S. Dubljevic, "Optimal control of axial dispersion tubular reactors with recycle: Addressing state-delay through transport PDEs," *The Canadian Journal of Chemical Engineering*, vol. 103, no. 8, pp. 3751–3766, 2025, ISSN: 0008-4034. DOI: 10.1002/cjce.25629.

[0] V. Havu and J. Malinen, "The Cayley transform as a time discretization scheme," *Numerical Functional Analysis and Optimization*, vol. 28, no. 7-8, pp. 825–851, 2007, ISSN: 0163-0563. DOI: 10.1080/01630560701493321.

[0] Q. Xu and S. Dubljevic, "Linear model predictive control for transport-reaction processes," *AIChE Journal*, vol. 63, no. 7, pp. 2644–2659, 2017, ISSN: 0001-1541. DOI: 10.1002/aic.15592.

[0] E. Hairer, M. Hochbruck, A. Iserles, and C. Lubich, "Geometric numerical integration," *Oberwolfach Reports*, vol. 3, no. 1, pp. 805–882, 2006, ISSN: 1660-8933. DOI: 10.4171/owr/2011/16.

Appendix A

Additional Example Figures

Each of the following pages will provide an example of a different figure configuration. In addition to the examples the code that generates the figure will be provided and explanations of what the different parts of the code do will be included. From all of the included information in this **??** it should be possible to even develop your own figures that potentially suit your needs best.
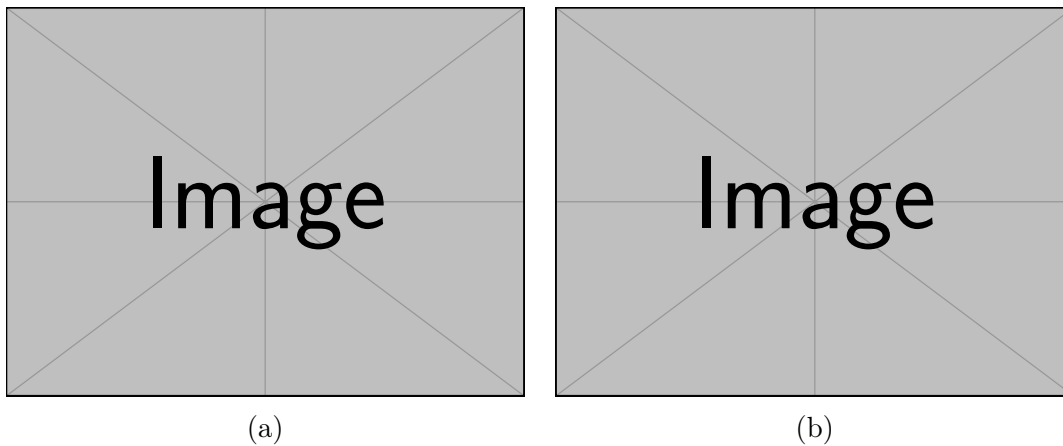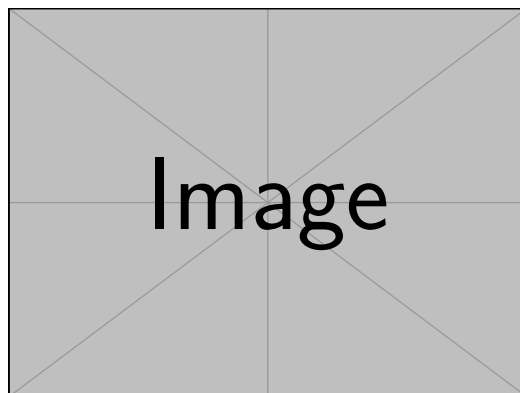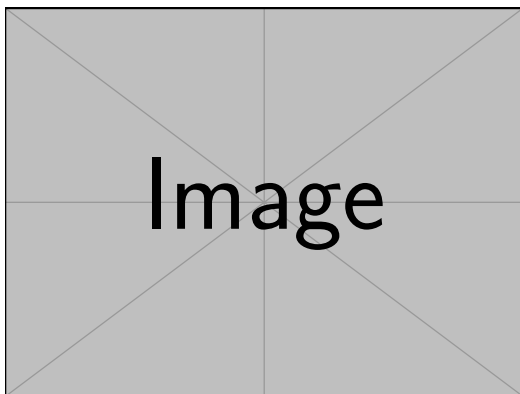
Figure A.1: This is an example of a double image figure.

```
\begin{figure}[H]
    \centering
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:doubleImage2:a}
    \end{subfigure}
    ~
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:doubleImage2:b}
    \end{subfigure}
    \caption{This is an example of a double image figure.}
    \label{fig:doubleImage2}
\end{figure}
```
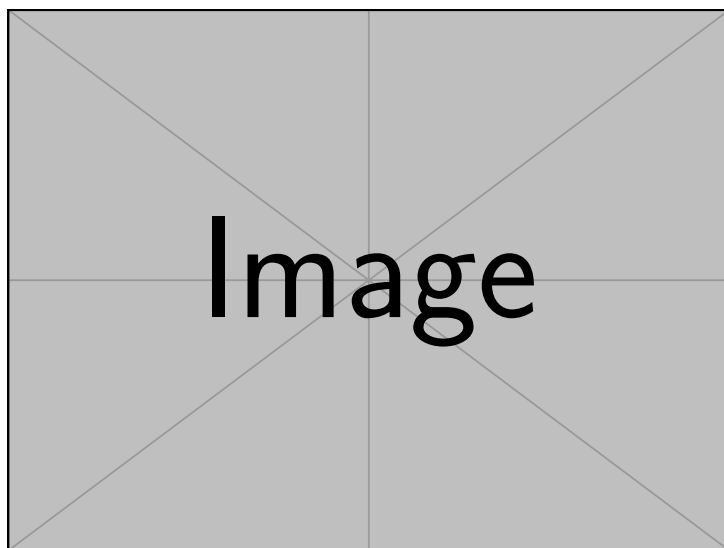
(a)

(b)
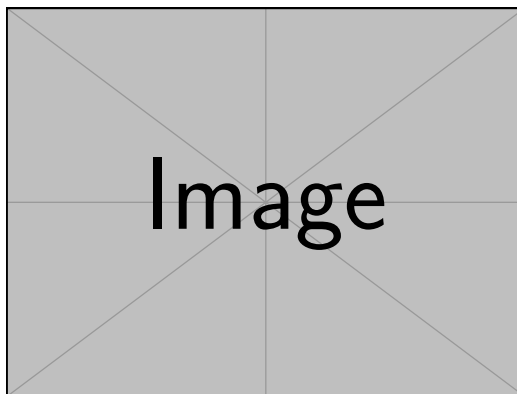
(c)

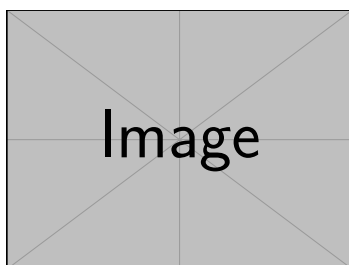Figure A.2: This is an example of a triple image figure.

```
\begin{figure}[H]
    \centering
    \hspace*{\fill}% Adds space to left of top image (prevents two images from going to top)
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:tripleImage1:a}
    \end{subfigure}
    \hspace*{\fill} % Adds space to right of top image (prevents two images from going to top)
    \par\vspace{1em}% Adds space between upper and lower images
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:tripleImage1:b}
    \end{subfigure}
    ~ % Adds space between the two lower figures
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:tripleImage1:c}
    \end{subfigure}
    \caption{This is an example of a triple image figure.}
    \label{fig:tripleImage1}
\end{figure}
```
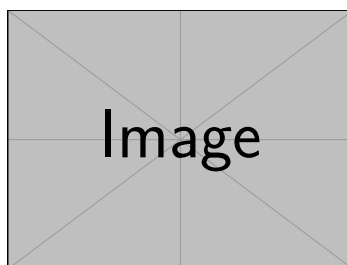
(a)

(b)   (c)

Figure A.3: This is a second example of a triple image figure.

```latex
\begin{figure}[H]
    \centering
    \hspace*{\fill}% Adds space to left of top image (prevents two images from going to top)
    \begin{subfigure}{0.90\linewidth+1em} % 0.9 = 0.45 + 0.45, and 1em is the width of ~
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:tripleImage2:a}
    \end{subfigure}
    \hspace*{\fill} % Adds space to right of top image (prevents two images from going to top)
    \par\vspace{1em}% Adds space between upper and lower images
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:tripleImage2:b}
    \end{subfigure}
    ~ % Adds space between the two lower figures
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:tripleImage2:c}
    \end{subfigure}
    \caption{This is a second example of a triple image figure.}
    \label{fig:tripleImage2}
\end{figure}
```
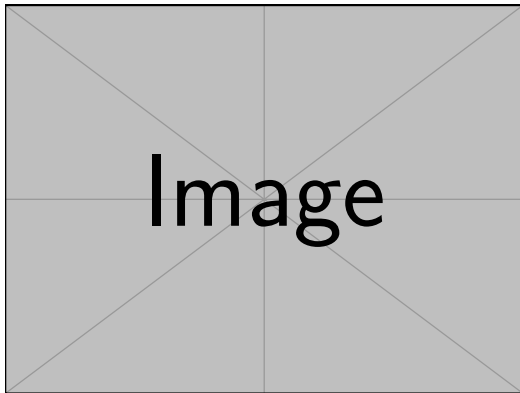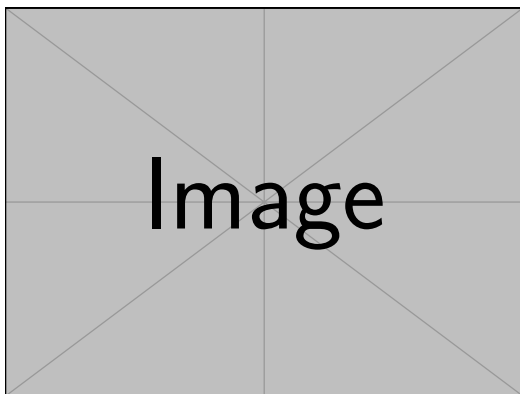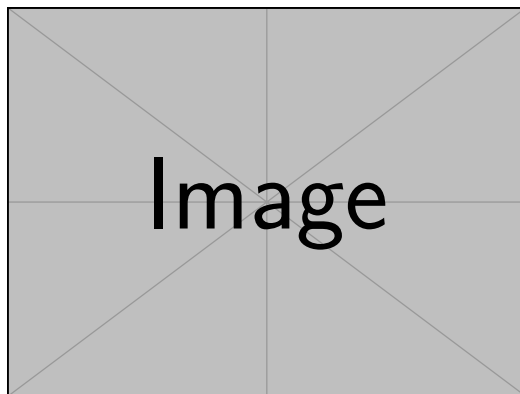
(a)
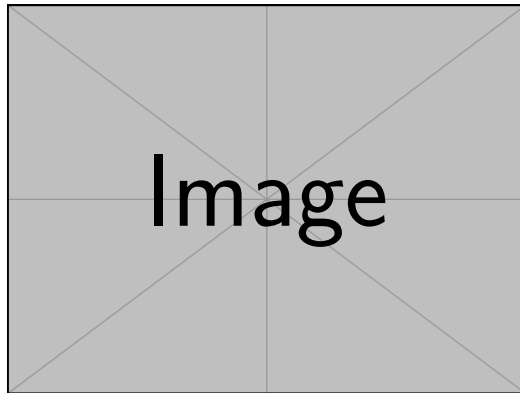
(b)

(c)

(d)

Figure A.4: This is an example of a quad image figure.

```latex
\begin{figure}[H]
    \centering
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:quadImage:a}
    \end{subfigure}
    ~ % Adds space between the two top figures
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:quadImage:b}
    \end{subfigure}
    \par\vspace{1em} % Adds space between upper and lower images
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:quadImage:c}
    \end{subfigure}
    ~ % Adds space between the two lower figures
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{example-image}
        \caption{} % Leave blank for just letter
        \label{fig:quadImage:d}
    \end{subfigure}
    \caption{This is an example of a quad image figure.}
    \label{fig:quadImage}
\end{figure}
```

Appendix B

Additional Example Tables

B.1 Section 1

Nullam eleifend justo in nisl. In hac habitasse platea dictumst. Morbi nonummy. Aliquam ut felis. In velit leo, dictum vitae, posuere id, vulputate nec, ante. Maecenas vitae pede nec dui dignissim suscipit. Morbi magna. Vestibulum id purus eget velit laoreet laoreet. Praesent sed leo vel nibh convallis blandit. Ut rutrum. Donec nibh. Donec interdum. Fusce sed pede sit amet elit rhoncus ultrices. Nullam at enim vitae pede vehicula iaculis.

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Aenean nonummy turpis id odio. Integer euismod imperdiet turpis. Ut nec leo nec diam imperdiet lacinia. Etiam eget lacus eget mi ultricies posuere. In placerat tristique tortor. Sed porta vestibulum metus. Nulla iaculis sollicitudin pede. Fusce luctus tellus in dolor. Curabitur auctor velit a sem. Morbi sapien. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Donec adipiscing urna vehicula nunc. Sed ornare leo in leo. In rhoncus leo ut dui. Aenean dolor quam, volutpat nec, fringilla id, consectetuer vel, pede.

Nulla malesuada risus ut urna. Aenean pretium velit sit amet metus. Duis iaculis. In hac habitasse platea dictumst. Nullam molestie turpis eget nisl. Duis a massa id

pede dapibus ultricies. Sed eu leo. In at mauris sit amet tortor bibendum varius. Phasellus justo risus, posuere in, sagittis ac, varius vel, tortor. Quisque id enim. Phasellus consequat, libero pretium nonummy fringilla, tortor lacus vestibulum nunc, ut rhoncus ligula neque id justo. Nullam accumsan euismod nunc. Proin vitae ipsum ac metus dictum tempus. Nam ut wisi. Quisque tortor felis, interdum ac, sodales a, semper a, sem. Curabitur in velit sit amet dui tristique sodales. Vivamus mauris pede, lacinia eget, pellentesque quis, scelerisque eu, est. Aliquam risus. Quisque bibendum pede eu dolor.

B.2   Section 2

Fusce suscipit cursus sem. Vivamus risus mi, egestas ac, imperdiet varius, faucibus quis, leo. Aenean tincidunt. Donec suscipit. Cras id justo quis nibh scelerisque dignissim. Aliquam sagittis elementum dolor. Aenean consectetuer justo in pede. Curabitur ullamcorper ligula nec orci. Aliquam purus turpis, aliquam id, ornare vitae, porttitor non, wisi. Maecenas luctus porta lorem. Donec vitae ligula eu ante pretium varius. Proin tortor metus, convallis et, hendrerit non, scelerisque in, urna. Cras quis libero eu ligula bibendum tempor. Vivamus tellus quam, malesuada eu, tempus sed, tempor sed, velit. Donec lacinia auctor libero.

Appendix C

Including Code Listings

This appendix provides guidelines for including code listings in your thesis. Code listings are often used to demonstrate algorithms, data processing scripts, or other relevant programming content. Proper formatting ensures that code is both readable and aesthetically pleasing.

C.1   Using the `listings` Package

The `listings` package is a powerful tool for displaying code in LaTeX. It supports syntax highlighting for a wide variety of programming languages and offers many customization options.

C.1.1   Basic Usage

To include a simple code listing, you can use the following command:

```
\begin{lstlisting}[language=Python]
    # Your code here
    print("Hello, world!")
\end{lstlisting}
```

The `language` option specifies the programming language, which enables syntax highlighting. Replace `Python` with the appropriate language for your code.

C.1.2 Customizing Listings

The `listings` package allows for extensive customization. You can adjust the appearance of your code by setting options such as `frame`, `backgroundcolor`, `keywordstyle`, and more.

Here is an example of how to customize your code listing:

```
\lstset{
    language=Python,
    frame=single,
    backgroundcolor=\color{gray!10},
    keywordstyle=\color{blue}\bfseries,
    commentstyle=\color{green},
    stringstyle=\color{red},
    basicstyle=\ttfamily,
    breaklines=true
}
```

This configuration adds a single-line frame around the code, sets a light gray background, and defines styles for keywords, comments, and strings.

Listing C.1: This is a caption for the inserted code

```
function [outputs] = functionName(inputs)
  %{
    This is a Comment Block
    That
    can
    span
    multiple
    lines.
  %}

  % This is a regular comment
  a = 1 + 2 * sin(angle);
  b = 'This is a String';
```

Listing C.2: This is a caption for the inserted code

```
#include <iostream>
using namespace std;
/* This function adds two integer values
 * and returns the result
 */

int sum(int num1, int num2){
    int num3 = num1 + num2; return num3;
}

void main(){
```

```
    //Calling the function
    cout << 'The sum is:' << sum(1,99);
}
```

C.2 Advanced Features

C.2.1 Including External Files

The `listings` package allows you to include code from external files. This is particularly useful if you have long code files that you want to reference directly.

```
\lstinputlisting[language=Python]{path/to/your/code.py}
```

Replace `path/to/your/code.py` with the actual path to your file. Or you can use the `\addmedia{./99_Inclusions/}` and `\addcode{Code/}` commands to define the location for code files, then you can just use `\insertcode{filename.ext}` command instead of the full path. You can customize the display in the same way as inline listings.

C.2.2 Handling Special Characters

If your code contains special characters (*e.g.*, #, %, $), you may need to escape them or use the `literate` option to ensure proper display.

```
\lstset{
    literate={~} {$\sim$}{1}
}
```

This command, for example, replaces the tilde symbol with the appropriate LaTeX command.

C.3 Line Breaks in Long Code Lines

To automatically break long lines of code, use the `breaklines=true` option as shown in the earlier examples. This prevents code from running off the page and maintains readability.

C.4 Conclusion

Including well-formatted code listings in your thesis can enhance the clarity of your work and demonstrate your technical skills. By following the guidelines in this appendix, you can ensure that your code is presented professionally.

Appendix D

Including PDFs

⚠ WARNING **While it is possible to have horizontal pages with the page numbers centered on the bottom long edge, I *DO NOT* recommended it. This is because, while it looks okay in a digital format, this is not suitable for printing... this would print page numbers on the side of the page rather than consistently on the bottom or in the heading.**

The package that is used to include PDF's is `pdfpages`. This provide the main command `\includepdf` that can be used to include a PDF. Further, this package provides another command, `\includepdfset`, that can be used in the start of the document to pre-set some of the default values. This class file detects the presence of this package and invokes this command as follows:

```
\includepdfset{pages=-,scale=0.85,pagecommand=\
   thispagestyle{STYLE}}
```

Where `pages=-` defaults to including all pages from the PDF and `scale=0.85` scales the inserted PDF to 85% of its original size, so that the documents fit within the page's margins, headers, and footers.

Further, the option `pagecommand=\cmd {thispagestyle}\mopt {STYLE}` sets the pagestyle for the PDF pages, where, STYLE is replaced with the current pagestyle[1]

---

[1]To change the pagestyle of this one can add or remove the class option `fancyheaders`.

that is in use.

D.1   How to Insert a Portrait PDF

To insert a portrait-oriented PDF into your LaTeX document, you can use the `pdfpages` package, which provides a convenient way to include external PDF files. The following code snippet demonstrates how to include a portrait PDF with the specified options:

```
\includepdf{./99_Inclusions/PDFs/examplePDF}
```

This is an Example PDF that is Portrait

This is the second page

D.2 How to Insert a Landscape PDF

Inserting a landscape-oriented PDF is similarly straightforward using the `pdfpages` package. The code snippet below demonstrates how to include a landscape PDF:

```
\includepdf[landscape=true]{./99_Inclusions/PDFs/
   landscapePDF}
```

Here, `landscape` sets the orientation to landscape. This configuration ensures that your landscape PDF is correctly oriented and properly sized within your document.

This is an Example PDF that is Landscape

This is the second page

Appendix E

Math Lettering

Table E.1: Math Mode Greek Letters

Command	Output	Command	Output	Command	Output
\alpha	α	\beta	β	\gamma	γ
\delta	δ	\epsilon	ϵ	\zeta	ζ
\eta	η	\theta	θ	\iota	ι
\kappa	κ	\lambda	λ	\mu	μ
\nu	ν	\xi	ξ	o	o
\pi	π	\rho	ρ	\sigma	σ
\tau	τ	\upsilon	υ	\phi	ϕ
\chi	χ	\psi	ψ	\omega	ω
A	A	B	B	\Gamma	Γ
\Delta	Δ	E	E	Z	Z
H	H	\Theta	Θ	I	I
K	K	\Lambda	Λ	M	M
N	N	\Xi	Ξ	O	O
\Pi	Π	P	P	\Sigma	Σ
T	T	\Upsilon	Υ	\Phi	Φ
X	X	\Psi	Ψ	\Omega	Ω

Table E.2: Blackboard Bold Letters

Command	Output	Command	Output	Command	Output
\mathbb\mopt{A}	$\mathbb{A}$	\mathbb\mopt{B}	$\mathbb{B}$	\mathbb\mopt{C}	$\mathbb{C}$
\mathbb\mopt{D}	$\mathbb{D}$	\mathbb\mopt{E}	$\mathbb{E}$	\mathbb\mopt{F}	$\mathbb{F}$
\mathbb\mopt{G}	$\mathbb{G}$	\mathbb\mopt{H}	$\mathbb{H}$	\mathbb\mopt{I}	$\mathbb{I}$
\mathbb\mopt{J}	$\mathbb{J}$	\mathbb\mopt{K}	$\mathbb{K}$	\mathbb\mopt{L}	$\mathbb{L}$
\mathbb\mopt{M}	$\mathbb{M}$	\mathbb\mopt{N}	$\mathbb{N}$	\mathbb\mopt{O}	$\mathbb{O}$
\mathbb\mopt{P}	$\mathbb{P}$	\mathbb\mopt{Q}	$\mathbb{Q}$	\mathbb\mopt{R}	$\mathbb{R}$
\mathbb\mopt{S}	$\mathbb{S}$	\mathbb\mopt{T}	$\mathbb{T}$	\mathbb\mopt{U}	$\mathbb{U}$
\mathbb\mopt{V}	$\mathbb{V}$	\mathbb\mopt{W}	$\mathbb{W}$	\mathbb\mopt{X}	$\mathbb{X}$
\mathbb\mopt{Y}	$\mathbb{Y}$	\mathbb\mopt{Z}	$\mathbb{Z}$		

Table E.3: Calligraphic Letters

Command	Output	Command	Output	Command	Output
\mathcal\mopt{A}	$\mathcal{A}$	\mathcal\mopt{B}	$\mathcal{B}$	\mathcal\mopt{C}	$\mathcal{C}$
\mathcal\mopt{D}	$\mathcal{D}$	\mathcal\mopt{E}	$\mathcal{E}$	\mathcal\mopt{F}	$\mathcal{F}$
\mathcal\mopt{G}	$\mathcal{G}$	\mathcal\mopt{H}	$\mathcal{H}$	\mathcal\mopt{I}	$\mathcal{I}$
\mathcal\mopt{J}	$\mathcal{J}$	\mathcal\mopt{K}	$\mathcal{K}$	\mathcal\mopt{L}	$\mathcal{L}$
\mathcal\mopt{M}	$\mathcal{M}$	\mathcal\mopt{N}	$\mathcal{N}$	\mathcal\mopt{O}	$\mathcal{O}$
\mathcal\mopt{P}	$\mathcal{P}$	\mathcal\mopt{Q}	$\mathcal{Q}$	\mathcal\mopt{R}	$\mathcal{R}$
\mathcal\mopt{S}	$\mathcal{S}$	\mathcal\mopt{T}	$\mathcal{T}$	\mathcal\mopt{U}	$\mathcal{U}$
\mathcal\mopt{V}	$\mathcal{V}$	\mathcal\mopt{W}	$\mathcal{W}$	\mathcal\mopt{X}	$\mathcal{X}$
\mathcal\mopt{Y}	$\mathcal{Y}$	\mathcal\mopt{Z}	$\mathcal{Z}$		

Table E.4: Fraktur Letters

Command	Output	Command	Output	Command	Output
\mathfrak\mopt {a}	𝔞	\mathfrak\mopt {b}	𝔟	\mathfrak\mopt {c}	𝔠
\mathfrak\mopt {d}	𝔡	\mathfrak\mopt {e}	𝔢	\mathfrak\mopt {f}	𝔣
\mathfrak\mopt {g}	𝔤	\mathfrak\mopt {h}	𝔥	\mathfrak\mopt {i}	𝔦
\mathfrak\mopt {j}	𝔧	\mathfrak\mopt {k}	𝔨	\mathfrak\mopt {l}	𝔩
\mathfrak\mopt {m}	𝔪	\mathfrak\mopt {n}	𝔫	\mathfrak\mopt {o}	𝔬
\mathfrak\mopt {p}	𝔭	\mathfrak\mopt {q}	𝔮	\mathfrak\mopt {r}	𝔯
\mathfrak\mopt {s}	𝔰	\mathfrak\mopt {t}	𝔱	\mathfrak\mopt {u}	𝔲
\mathfrak\mopt {v}	𝔳	\mathfrak\mopt {w}	𝔴	\mathfrak\mopt {x}	𝔵
\mathfrak\mopt {y}	𝔶	\mathfrak\mopt {z}	𝔷		
\mathfrak\mopt {A}	𝔄	\mathfrak\mopt {B}	𝔅	\mathfrak\mopt {C}	ℭ
\mathfrak\mopt {D}	𝔇	\mathfrak\mopt {E}	𝔈	\mathfrak\mopt {F}	𝔉
\mathfrak\mopt {G}	𝔊	\mathfrak\mopt {H}	ℌ	\mathfrak\mopt {I}	ℑ
\mathfrak\mopt {J}	𝔍	\mathfrak\mopt {K}	𝔎	\mathfrak\mopt {L}	𝔏
\mathfrak\mopt {M}	𝔐	\mathfrak\mopt {N}	𝔑	\mathfrak\mopt {O}	𝔒
\mathfrak\mopt {P}	𝔓	\mathfrak\mopt {Q}	𝔔	\mathfrak\mopt {R}	ℜ
\mathfrak\mopt {S}	𝔖	\mathfrak\mopt {T}	𝔗	\mathfrak\mopt {U}	𝔘
\mathfrak\mopt {V}	𝔙	\mathfrak\mopt {W}	𝔚	\mathfrak\mopt {X}	𝔛
\mathfrak\mopt {Y}	𝔜	\mathfrak\mopt {Z}	ℨ		