An NTNU Thesis typst template

Anders Andersen
Bjørn Bjørnsen

27/08/2024

Abstract

The ntnu-thesis template is a typst port of the ntnuthesis LaTeX class. It can be used for theses at all levels – bachelor, master and PhD – and is available in English (British and American) and Norwegian (Bokmål and Nynorsk). This document is ment to serve (i) as a description of the document class, (ii) as an example of how to use it, and (iii) as a thesis template.

Sammendrag

Malen `ntnu-thesis` er en typst-oversettelse av LaTeX-klassen `ntnuthesis`. Den er tilrettelagt for avhandlinger på alle nivåer – bachelor, master og PhD – og er tilgjengelig på både norsk (bokmål og nynorsk) og engelsk (britisk og amerikansk). Dette dokumentet er ment å tjene (i) som en beskrivelse av dokumentklassen, (ii) som et eksempel på bruken av den, og (iii) som en mal for avhandlingen.

Contents

Figures

Tables

Code listings

Chapter 1

Introduction

The original `ntnuthesis` template was created by the CoPCSE[1] as a template applicable for theses at all study levels. It is closely based on the standard LaTeX `report` document class as well as previous thesis templates. This typst port aims to replicate the look of the LaTeX template in typst.

The purpose of the present document is threefold. It should serve (i) as a description of the document class, (ii) as an example of how to use it, and (iii) as a thesis template.

[1] https://www.ntnu.no/wiki/display/copcse/Community+of+Practice+in+Computer+Science+Education+Home

Chapter 2
Using the Template

2.1 Thesis Setup

The document class is initialized by calling `#show: ntnu-thesis.with()` at the beginning of your `.typ` file. Currently it only supports english. The `ntnu-thesis` function has a number of options you can set, most of which will be described in this document. The rest will be documented in this templates repository.

The titlepage at the beginning of this document is a placeholder to be used when writing the thesis. This should be removed before handing in the thesis, by settting `titlepage: false` . Instead the official NTNU titlepage for the corresponding thesis type should be added as described on Innsida.[2]

2.2 Title, Author, and Date

The title of your thesis should be set by changing the `title` parameter of the template. The title will appear on the titlepage as well as in the running header of the even numbered pages. If the title is too long for the header, you can use `shorttitle` to set a version for the header.

The authors should be listed with full names in the `authors` parameter. This is an array, with multiple authors separated by a comma.

Use `date` to set the date of the document. It will only appear on the temporary title page. To keep track of temporary versions, it can be a good idea to use `date: datetime.today()` while working on the thesis.

[2] see https://innsida.ntnu.no/wiki/-/wiki/English/Finalizing+the+bachelor+and+master+thesis for bachelor and master, and https://innsida.ntnu.no/wiki/-/wiki/English/Printing+your+thesis for PhD.

2.3 Page Layout

The document class is designed to work with twosided printing. This means that all chapters start on odd (right hand) pages, and that blank pages are inserted where needed to make sure this happens. However, since the theses are very often read on displays, the margins are kept the same on even and odd pages in order to avoid that the page is jumping back and forth upon reading.

By default this is turned off. You can turn it on by setting `chapters-on-odd: false` at the top of the file.

2.4 Structuring Elements

The standard typst headings are supported, and are set using =.

2.4.1 This is a level 3 heading

2.4.1.1 This is level 4 heading

2.4.1.1.1 This is a level 5 heading

Headings up to level 3 will be included in the table of contents, whereas the lower level structuring elements will not appear there. Don’t use too many levels of headings; how many are appropriate, will depend on the size of the document. Also, don’t use headings too frequently.

Make sure that the chapter and section headings are correctly capitalised depending on the language of the thesis, e.g., ‘*Correct Capitalisation of Titles in English*’ vs. ‘*Korrekt staving av titler på norsk*’.

Simple paragraphs are the lowest structuring elements and should be used the most. They are made by leaving one (or more) blank line(s) in the `.typ` file. In the typeset document they will appear indented and with no vertical space between them.

2.5 Lists

Numbered and unnumbered lists are used just as in regular typst, but are typeset somewhat more densely and with other labels. Unnumbered list:

- first item

- second item
  - first subitem
  - second subitem
    - first subsubitem
    - second subsubitem
- last item

Numbered list:

1. first item
2. second item
   a. first subitem
   b. second subitem
      i. first subsubitem
      ii. second subsubitem
3. last item

2.6 Figures

Figures are added using `#figure()` . An example is shown in 2.1. By default figures are placed in the flow, exactly where it was specified. To change this set the option to either `top` , `bottom` , or `auto` . To add an image, use `#image()` and set the `height` or `width` to include the graphics file. If the caption consists of a single sentence fragment (incomplete sentence), it should not be punctuated.

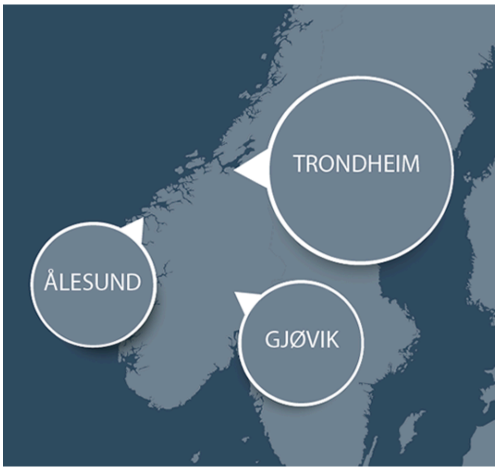
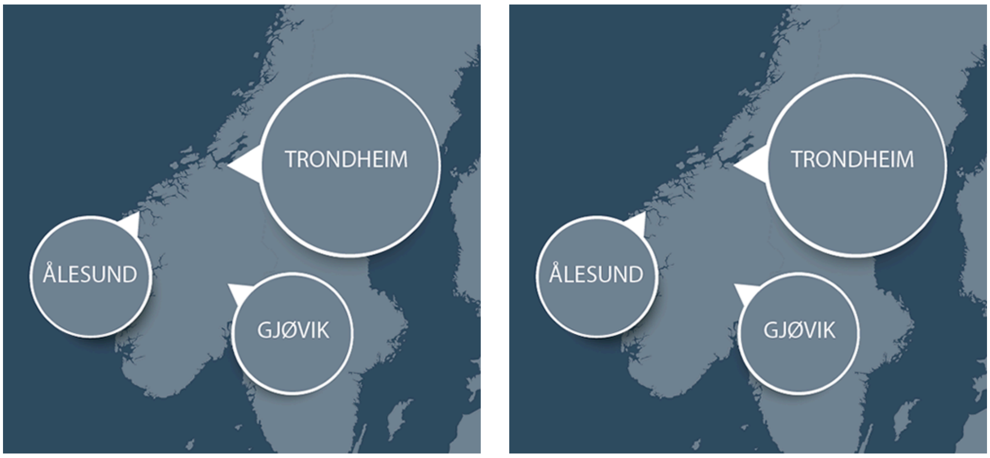


Figure 2.1: The map shows the three main campuses of NTNU.

For figures compsed of several sub-figures, the `subpar` module has been used. See 2.4 with [sfig:a] for an example.



(a) First sub-figure (b) Second sub-figure

Figure 2.2: A figure composed of two sub-figures. It has a long caption in order to demonstrate how that is typeset.

2.7 Tables

Tables are added using `#table()`, wrapped in a `#figure()` to allow referencing. An example is given in Table 2.1. If the caption consists of a single sentence fragment (incomplete sentence), it should not be punctuated.

Table 2.1: A simple, manually formatted example table

age	**IQ**
10	110
20	120
30	145
40	120
50	100

Tables can also be automatically generated from CSV files[3].

[3] https://typst.app/docs/reference/data-loading/csv/

2.8 Listings

Code listings are are also wrapped in a #figure() . Code listings are defined by using three `backticks` . The programming language can also be provided. See the typst documentation for details. The code is set with the monospace font, and the font size is reduced to allow for code lines up to at least 60 characters without causing line breaks. If the caption consists of a single sentence fragment (incomplete sentence), it should not be punctuated.

Listing 2.1: Python code in typst

```python
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 1)
y = np.sin(2 * np.pi * x)

plt.plot(x, y)
plt.show()
```

Listing 2.2: C++ code in typst

```cpp
#include <iostream>
using namespace std;

int main()
{
  cout << "Hello, World!" << endl;
  return 0;
}
```

2.9 Equations

Equations are typeset as normally in typst. It is common to consider equations part of the surrounding sentences, and include punctuation in the equations accordingly, e.g.,

$$f(x) = \int_1^x \frac{1}{y}\, dy = \ln x\,. \tag{2.1}$$

For more advanced symbols like, e.g., ∇, $\frac{\partial x}{\partial y}$, the physica module is preloaded. As you can see, the simple math syntax makes typst very easy to use.

2.10 Fonts

Charter at 11pt with the has been selected as the main font for the thesis template. For code examples, the monospaced font should be used – for this, a scaled version of the DejaVu Sans Mono to match the main font is preselected.

2.11 Cross References

Cross references are inserted using `=` in typst. For examples on usage, see Chapter 2.11 in Chapter 2, Table 2.1 Figure 2.1, Equation (2.1), Listing 2.2 and Appendix A.

2.12 Bibliography

The bibliography is typeset as in standard typst. It is added in the initializing function as such: `bibliography: bibliography("thesis.bib")`. With this setup, using `@` will give a number only [1], and `#cite(, form: "prose")` will give author and number like this: K. K. Landes [1].

2.13 Appendices

Additional material that does not fit in the main thesis but may still be relevant to share, e.g., raw data from experiments and surveys, code listings, additional plots, pre-project reports, project agreements, contracts, logs etc., can be put in appendices. Simply issue the command `#show: appendix` in the main `.typst` file, and then the following chapters become appendices. See Appendix A for an example.

Chapter 3
Thesis Structure

The following is lifted more or less directly from the original template.

The structure of the thesis, i.e., which chapters and other document elements that should be included, depends on several factors such as the study level (bachelor, master, PhD), the type of project it describes (development, research, investigation, consulting), and the diversity (narrow, broad). Thus, there are no exact rules for how to do it, so whatever follows should be taken as guidelines only.

A thesis, like any book or report, can typically be divided into three parts: front matter, body matter, and back matter. Of these, the body matter is by far the most important one, and also the one that varies the most between thesis types.

3.1 Front Matter

The front matter is everything that comes before the main part of the thesis. It is common to use roman page numbers for this part to indicate this. The minimum required front matter consists of a title page, abstract(s), and a table of contents. A more complete front matter, in a typical order, is as follows.

Title page:

The title page should, at minimum, include the thesis title, authors and a date. A more complete title page would also include the name of the study programme, and possibly the thesis supervisor(s). See 2.1.

Abstracts:

The abstract should be an extremely condensed version of the thesis. Think one sentence with the main message from each of the chapters of the body matter as a starting point. K. K. Landes [1] have given some very nice instructions on how to write a good abstract. A thesis from a Norwegian Univeristy should contain abstracts in both Norwegian and English irrespectively of the thesis language (typically with the thesis language coming first).

Dedication:

If you wish to dedicate the thesis to someone (increasingly common with increasing study level), you may add a separate page with a dedication here. Since a dedication is a personal statement, no template is given. Design it according to your preference.

Acknowledgements:

If there is someone who deserves a 'thank you', you may add acknowledgements here. If so, make it an unnumbered chapter.

Table of contents:

A table of contents should always be present in a document at the size of a thesis. It is generated automatically using the `outline()` command. The one generated by this document class also contains the front matter and unnumbered chapters.

List of figures:

If the thesis contains many figures that the reader might want to refer back to, a list of figures can be included here. It is generated using `outline()` .

List of tables:

If the thesis contains many tables that the reader might want to refer back to, a list of tables can be included here. It is generated using `outline()` .

List of code listings:

If the thesis contains many code listings that the reader might want to refer back to, a list of code listings can be included here. It is generated using `outline()` .

Other lists:

If there are other list you would like to include, this would be a good place. Examples could be lists of definitions, theorems, nomenclature, abbreviations, glossary etc.

Preface or Foreword:

A preface or foreword is a good place to make other personal statements that do not fit whithin the body matter. This could be information about the circumstances of the thesis, your motivation for choosing it, or possibly information about an employer or an external company for which it has been written. Add this in the initializing function of this template.

3.2 Body Matter

The body matter consists of the main chapters of the thesis. It starts the Arabic page numbering with page 1. There is a great diversity in the structure chosen for different thesis types. Common to almost all is that the first chapter is an introduction, and that the last one is a conclusion followed by the bibliography.

3.2.1 Development Project

For many bachelor and some master projects in computer science, the main task is to develop something, typically a software prototype, for an 'employer' (e.g., an external company or a research group). A thesis describing such a project is typically structured as a software development report whith more or less the following chapters:

Introduction:

The introduction of the thesis should take the reader all the way from the big picture and context of the project to the concrete task that has been solved in the thesis. A nice skeleton for a good introduction was given by J. F. Claerbout [2]: *review–claim–agenda*. In the review part, the background of the project is covered. This leads up to your claim, which is typically that some entity (software, device) or knowledge (research questions) is missing and sorely needed. The agenda part briefly summarises how your thesis contributes.

Requirements:

The requirements chapter should lead up to a concrete description of both the functional and non-functional requirements for whatever is to be developed at both a high level (use cases) and lower levels (low level use cases, requirements). If a classical waterfall development process is followed, this chapter is the product of the requirement phase. If a more agile model like, e.g., SCRUM is followed, the requirements will appear through the project as, e.g., the user stories developed in the sprint planning meetings.

Technical design:

The technical design chapter describes the big picture of the chosen solution. For a software development project, this would typically contain the system arcitechture (client-server, cloud, databases, networking, services etc.); both how it was solved, and, more importantly, why this architecture was chosen.

Development Process:

In this chapter, you should describe the process that was followed. It should cover the process model, why it was chosen, and how it was implemented, including tools for project management, documentation etc. Depending on how you write the other chapters, there may be good reasons to place this chapters somewhere else in the thesis.

Implementation:

Here you should describe the more technical details of the solution. Which tools were used (programming languages, libraries, IDEs, APIs, frameworks, etc.). It is a good idea to give some code examples. If class diagrams, database models etc. were not presented in the technical design chapter, they can be included here.

Deployment:

This chapter should describe how your solution can be deployed on the employer's system. It should include technical details on how to set it up, as well as discussions on choices made concerning scalability, maintenance, etc.

Testing and user feedback:

This chapter should describe how the system was tested during and after development. This would cover everything from unit testing to user testing; black-box vs. white-box; how it was done, what was learned from the testing, and what impact it had on the product and process.

Discussion:

Here you should discuss all aspect of your thesis and project. How did the process work? Which choices did you make, and what did you learn from it? What were the pros and cons? What would you have done differently if you were to undertake the same project over again, both in terms of process and product? What are the societal consequences of your work?

Conclusion:

The conclusion chapter is usually quite short – a paragraph or two – mainly summarising what was achieved in the project. It should answer the *claim* part of the introduction. It should also say something about what comes next ('future work').

Bibliography:

The bibliography should be a list of quality-assured peer-reviewed published material that you have used throughout the work with your thesis. All items in the bibliography should be referenced in the text. The references should be correctly formatted depending on their type (book, journal article, conference publication, thesis etc.). The bibliography should not contain links to arbitrary dynamic web pages where the content is subject to change at any point of time. Such links, if necessary, should rather be included as footnotes throughout the document. The main point of the bibliography is to back up your claims with quality-assured material that future readers will actually be able to retrieve years ahead.

3.2.2 Research Project

For many master and some bachelor projects in computer science, the main task is to gain knew knowledge about something. A thesis describing such a project is typically structed as an extended form of a scientific paper, following the so-called IMRaD (Introduction, Method, Results, and Discussion) model:

Introduction:

See 3.2.1.

Background:

Research projects should always be based on previous research on the same and/or related topics. This should be described as a background to the thesis with adequate bibliographical references. If the material needed is too voluminous to fit nicely in the review part of the introduction, it can be presented in a separate background chapter.

Method:

The method chapter should describe in detail which activities you undertake to answer the research questions presented in the introduction, and why they were chosen. This includes detailed descriptions of experiments, surveys, computations, data analysis, statistical tests etc.

Results:

The results chapter should simply present the results of applying the methods presented in the method chapter without further ado. This chapter will typically contain many graphs, tables, etc. Sometimes it is natural to discuss the results as they are presented, combining them into a 'Results and Discussion' chapter, but more often they are kept separate.

Discussion:

See 3.2.1.

Conclusion:

See 3.2.1.

Bibliography:

See 3.2.1.

3.2.3 Monograph PhD Thesis

Traditionally, it has been common to structure a PhD thesis as a single book – a *monograph*. If the thesis is in the form of one single coherent research project, it can be structured along the lines of 3.2.2. However, for such a big work that a PhD thesis constitutes, the tasks undertaken are often more diverse, and thus more naturally split into several smaller research projects as follows:

Introduction:

The introduction would serve the same purpose as for a smaller research project described in 3.2.1, but would normally be somewhat more extensive. The *agenda* part should inform the reader about the structure of the rest of the document, since this may vary significantly between theses.

Background:

Where as background chapters are not necessarily needed in smaller works, they are almost always need in PhD thesis. They may even be split into several chapters if there are significantly different topics to cover. See 3.2.2.

Main chapters:

Each main chapter can be structured more or less like a scientific paper. Depending on how much is contained in the introduction and background sections, the individual introduction and background sections can be significantly reduced or even omitted completely.

- (Introduction)

- (Background)
- Method
- Results
- Discussion
- Conclusion

Discussion:

In addition to the discussions within each of the individual chapters, the contribution of the thesis *as a whole* should be thoroughly discussed here.

Conclusion:

In addition to the conclusions of each of the individual chapters, the overall conclusion of the thesis, and how the different parts contribute to it, should be presented here. The conclusion should answer to the research questions set out in the main introduction. See also 3.2.1.

Bibliography:

See 3.2.1.

3.2.4 Compiled PhD Thesis

Instead of writing up the PhD thesis as a monograph, compiled PhD theses (also known as stapler theses, sandwich theses, integrated theses, PhD by published work) consisting of reproductions of already published research papers are becoming increasingly common. At least some of the papers should already have been accepted for publication at the time of submission of the thesis, and thus have been through a real quality control by peer review.

Introduction:

See 3.2.3.

Background:

See 3.2.3.

Main contributions:

This chapter should sum up *and integrate* the contribution of the thesis as a whole. It should not merely be a listing of the abstracts of the individual papers – they are already available in the attached papers, and, as such, not needed here.

Discussion:

See 3.2.3.

Conclusion:

See 3.2.3.

Bibliography:

See 3.2.1.

Paper I:

First included paper with main contributions. It can be included verbatim as a PDF. The publishers PDF should be used if the copyright permits it. This should be checked with the SHERPA/RoMEO database[4] or with the publisher. Even when it is no general permission by the publisher, you may write and ask for one.

Paper II:

etc.

3.3 Back Matter

Material that does not fit elsewhere, but that you would still like to share with the readers, can be put in appendices. See 5.

[4] http://sherpa.ac.uk/romeo/index.php

Chapter 4
Conclusion

You definitely should use the `ntnu-thesis` typst template for your thesis.

Appendix A
Additional Material

Additional material that does not fit in the main thesis but may still be relevant to share, e.g., raw data from experiments and surveys, code listings, additional plots, pre-project reports, project agreements, contracts, logs etc., can be put in appendices. Simply issue the command `#show: appendix` in the main `.typ` file, and make one chapter per appendix.

Bibliography

[1] K. K. Landes, "A Scrutiny of the Abstract," *Bulletin of the American Association of Petroleum Geologists*, vol. 35, no. 7, p. 1660–1661, 1951.

[2] J. F. Claerbout, "A Scrutiny of the Introduction," *The Leading Edge*, vol. 10, no. 1, pp. 39–41, 1991.