UPPSALA
UNIVERSITET

UPTEC XX XXXXX
Degree project X credits
February, 2026

# Your Fancy Title

Subtitle?

First name Last name

Master's Programme in X

UPPSALA
UNIVERSITET

Your Fancy Title

First name Last name

**Abstract** Abstract, always in English, about 10-20 lines. Do not use references, and do not use formulas if they can be avoided.

1. What is the problem/issue/subject?
2. How was the problem solved/attacked?
3. What are the results, how well was the problem solved?
4. How good are the results, how useful are they?

The abstract should be understandable without reading the whole report (and the rest of the report should be understandable without reading the abstract). You can reuse text/phrases from the Introduction.

**Faculty of Science and Technology**

**Uppsala University, Place of publication Uppsala**

Supervisor: X  Subject reader: Y
Examiner: Z

# Sammanfattning

Sammanfattning, alltid på svenska. Se till att det står samma saker i den svenska sammanfattningen och det engelska abstractet.

1. Vad är problemet, ämnet?
2. Hur angreps/löstes problemet?
3. Vad är resultaten, hur väl löstes problemet?
4. Hur bra blev resultaten, hur användbara är de?

Ca 10-20 rader. Använd inte referenser; ej heller formler om det går att undvika.

Sammanfattningen ska vara förståelig utan att läsa resten av rapporten, och resten av rapporten ska kunna läsas utan att läsa sammanfattningen. Det är helt OK att återanvända text från introduktionen.

# Acknowledgements

Acknowledgements are not always necessary, only sometimes: if you have received significant help/feedback/support from someone (beyond what can be expected, e.g., from a supervisor), if you have received special permission to use some material or equipment, or if you have received financial sponsorship (which, for example, could affect scientific impartiality). The section should be brief and concise, typically one paragraph.

**NOTE:** Check *Section B.4.2* for how to define and use acronyms.

# Acronyms

**BC** Boundary Condition

**DOF** Degree of Freedom

**MOL** Method of Lines

**PDE** Partial Differential Equation

**TGV** High speed train, from French *Train à haute vitesse*

# Contents

**NOTE:** The order and types of chapters/sections provided are just examples. You should decide on the best chapters and order for your work.

# 1 Introduction

Describe at least the same things as in the abstract, but in more detail — typically 1-2 pages. Save technical details for later, as the reader is not yet familiar with them.

1. What is the area you are working in? What is the problem, subject, context?
2. Why is the problem important/interesting to solve?
3. How was the problem approached/solved?
4. What are the results, how well was the problem solved?
5. How good are the results, how useful are they?

Remember to start the introduction with a sentence or, even better, a whole paragraph that "captures" the reader and motivates them to keep reading. Starting with "We have chosen to do a project about X" might be relevant for you, but it won't make anyone want to read further. Try to include some background/context and (preferably) motivation to keep reading.

Make sure to get to the point quickly — the reader doesn't want to go through two paragraphs of text before finding out what you plan to do in your project. For example, don't start by presenting all the ideas you didn't choose — the reader wants to know what you are going to do, not what you are not going to do.

The introduction should be understandable to, for example, a student in the year below you, and preferably to an even broader audience. Feel free to conclude with an overview of the entire report, briefly describing what the different sections are about. For example, *Section 2* describes the purpose, aims and motivation. *Section 3* outlines…

# 2 Purpose, Aims and Motivation

Here you essentially describe your problem formulation. This section should state the purpose, goals, and motivation for the project. These do not need to be separate subheadings.

**Purpose:** What is the project striving for? What is the overall goal, benefit, or effect of the project? (e.g., better health through dietary monitoring, easier planning of studies…)

**Goals:** What specific deliverables or actions will the project accomplish to bring us closer to the purpose?

**Motivation:** Why is your project important? Who is it important for, and which external stakeholders are there? How significant is the problem, what are the consequences if it isn't solved, and how beneficial would it be to solve it? What "gap" in the field are you addressing? Why is your solution better/different than others'?

Avoid discussing your internal motivation, as it is rarely of interest to the reader.

Make sure to convince the reader in this section that the problem exists, that it isn't solved, and that it is important to solve. The stronger the argumentation and motivation (with sources), the better.

1. Show that there is a problem.
2. Show that the problem is important to solve and needs solving.
3. Show that the problem is not already solved.

In this section, you can also start describing ethical aspects and sustainability aspects, this could be in subsections.

## 2.1 Delimitations

Here you describe what you have not done, i.e., how you have chosen to limit yourselves, and justify these limitations. This clarifies for the reader who may have had expectations you have not met.

# 3 Background

Here, you describe the background of your project, i.e., what leads up to your problem formulation. What is the area, environment, context, and background for the project (more detailed and deeper than in the introduction)? Describe the area (e.g., sound processing, study plans, visualization, autism…). Describe the client, if you have one (but not too detailed). Keep in mind that the background and the problem must be at a general academic level and not just related to a specific client.

Remember that the background can look further back — how was the problem solved before? Before computerization? Sometimes it is both important and interesting (but sometimes not).

After reading the background, it should be easy to understand the purpose/goals and why they are important, and it should be clear that you are well-versed in the subject.

# 4 Method

Here, you describe which methods/tools/techniques/approaches you used to solve the problem/answer the question. What methods did you concretely use to solve the problem? Which techniques/tools did you use?

Don't forget to justify your choice of methods. Are there several reasonable alternatives? Describe why you did not choose them (e.g., why your chosen method is better). Show that it is reasonable to use this particular approach. This also applies in cases where it is predetermined which technique you should use.

# 5 Results and Discussion

Here you first describe your results, what you have accomplished. How good did it turn out? Then you critically review your results. Why did it turn out the way it did? Were the results reasonable/good/bad/unexpected…? What could have been done differently? How do your results relate to similar works?

1. Show that the evaluation is reasonable.

2. Show that the evaluation, results, and analysis are scientific and engineering-based.

Relate to the goals and purposes in *Section 2*, and use any relevant theory from *Section 4* to explain.

# 6 Conclusions

Here you summarize and restate your contribution (the results of your project) and explain its significance and usage. What was important/new/interesting? (NOT in terms of what you learned, but for those reading the report)

Summarize the most important parts of the project, but not by copy-pasting; preferably with a new formulation. Try to place the most important parts in a larger perspective or context.

## 6.1 Future Work

Here you describe potential future developments of the thesis. Where is there potential for improvement and what can be built upon? What interesting extensions did you not have time for?

# 7 Declaration of generative AI use

Describe any use of generative AI in the creation of this work, including tools used and how they were utilized. This applies to both the project and the report. This is important for transparency and academic integrity.

# Bibliography

[1] E. Bell, A. Bryman, and B. Harley, *Business research methods*, Fifth edition. Oxford University Press, 2019.

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing,” *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004, doi: 10.1109/TDSC.2004.2.

[3] Uppsala University, “Almedalen.” Accessed: June 17, 2024. [Online]. Available: https://www.uu.se/

# A Appendix

Appendices can be useful for attachments such as surveys, larger code segments, etc.

Appendices should be placed after the reference list and should start on a new page. Use `#pagebreak()` to create a page break where the rest of the current page is blank. Then use `#set heading(numbering: "A.1.1.1 ")` to change heading to indicate that we are in the appendix and `#context counter(heading).update(0)` to reset the heading counter. Then use the usual `== Label <app:label>` for each appendix, which will be "numbered" A.1, A.2, A.3, etc. In this project using `<app:...>` in the label will replace the supplement with "Appendix" when refferenced.

To refer to an appendix, do the same as for a section.

# B Some Tips for Typst Usage

To **refer** to sections, figures, tables, etc., use `<sec:label>` or `#label("sec:label")` to "set a mark" in the text or figure, and `@sec:label` or `#ref(sec:label)` to refer to it, for example, Read more on *page 10* about naming sections, figures, etc. In Typst you can't label and refer to general text, but you can use headings for that, e.g.,

```typst
= Methods <sec:method>
// #label("sec:methods")
```

followed by

```typst
As mentioned in~@sec:methods...
```

To prevent references from appearing immediately after a **line break**, use a non-breaking space `like~this`, where the tilde character `~` creates a non-breaking space. This is also generally correct to use before numbers (and in large numbers in English, e.g., `100~000` for 100,000), and of course also before references to figures `@fig`.

To create a **paragraphbreak** it's best to use a blank line or `#parbreak()`.

To create a **page break** where the rest of the page becomes blank, use `#pagebreak()`

## B.1 Hayagriva for Bibliography and References

To manage bibliography (**references**) smoothly, use Hayagriva! Hayagriva is a new bibliography file format designed for use with Typst. Read more at here and use this as a guide for file format. An example bibliography file is provided as `ref_examples.yml`. Typst also has suport for Bib(La)Tex files, but Hayagriva is more powerful and has a nicer syntax.

```yaml
harry:
    type: Book
    title: Harry Potter and the Order of the Phoenix
    author: Rowling, J. K.
```

```
    volume: 5
    page-total: 768
    date: 2003-06-21

electronic:
    type: Web
    title: Ishkur's Guide to Electronic Music
    serial-number: v2.5
    author: Ishkur
    url:
        value: http://www.techno.org/electronic-music-guide/
        date: 2026-01-26
```

Multiple authors are presented as an array of strings.

```yaml
author: ["Omarova, Saule", "Steele, Graham"]
```

And some fields accept composite data such as the `url` field above where the date specifies access date.

## B.2 References

> **NOTE: important!** There are at least three purposes for the design of references and the reference list.
>
> 1. One should find the reference (from the text) in the reference list.
> 2. One should understand what is being referred to (what type of reference it is) so that one can evaluate it.
> 3. One should be able to find the reference in reality.
>
> Always strive to achieve all three.

Use numerical references (IEEE style [42]) or keyword-based [Lam86], not footnote style. The references are sorted alphabetically by author/

etc. in the reference list. In Typst, use `#bibliography{"refs.yml", style: "ieee"}`, see the main.tex.

References are written in direct connection to what prompted the reference (e.g., a statement or result), before any punctuation, and with a non-breaking space to the previous word. In Typst, `write~@Lam86` to get a "non-breaking space."

One should *not* write the references after a longer paragraph (as some seem to be taught to do, somewhere). This usually makes it unclear what is actually taken from, or supported by, the references. In some cases, you may want to provide a short summary of what an author writes in an article, etc., but just adding a reference at the end of the paragraph is not clear enough. It is much better and clearer to start the paragraph by writing something like "Lisa Lagom describes`~@lagom-bok` how X depends on Y and in her analysis, she shows in detail how the relationship looks…"

Repeating the same reference often in a paragraph (maybe after every sentence) makes it more difficult to read. Try to rewrite the paragraph so that it becomes clear that it is all based on the reference, which should preferably be used early. Example: "In a study by WHO`~@who`, the consequences of XYZ and the indirect risks that arise are described," and then the different consequences and risks can be discussed in the same paragraph without it becoming unclear.

When referencing "thick" items like books, it is appropriate to specify page numbers (as `@example:alma[pp. 211-214]`) which becomes *[1, pp. 211-214]*, but for "thinner" items, you only need to do this to specifically point out if you mean a particular part of the reference (maybe it describes three different ways to do X and you want to point to the 3rd, not the first two).

To refer to multiple things simultaneously, write multiple reference keys after eachother `@example:alma @example:dependability` which becomes *[1], [2]*.

Note that news articles (newspaper articles and the like) almost always have a publication date, which should be shown (e.g., in the `date` field).

Even if a reference has a URL to the actual text, it is not necessarily a web reference, but sometimes an article/book, etc., that happens to be available online. It should then be described as an article/book/etc. (but of course preferably with the URL) so that one can make a preliminary assessment of the reference already when reading the reference list.

Try to find authors and publication dates (year, month) even for web references, and **always** specify when they were accessed, as they can be updated at any time. This is done by setting the composit values

```
{value: X, date: Y}
```

on the `url` field. An example is *[3]* (see `refs.yml`).

## B.3 Formulas, Figures, Images, Code, Timelines

Formulas, figures, and equations must be described. This means, for example, that each symbol must be explained in the text.

Figures and tables that are “floats” in LaTeX are more deterministic in Typst and generally end up within a page of where you place them. Still let figures and tables end up where Typst thinks they should, and adjust the placement only in the final version and if it is really necessary.

Figure captions should describe what we see in the figure, not just what type of figure it is. Writing “System structure” or “The structure of our system” for a picture of the system structure is not sufficient. Help the reader understand by also (or instead) describing the content, e.g., “The green circles represent users, and components with shaded background are external. Input comes from the left, and output is delivered to the right.” It is therefore *not* enough to describe the figure in the running text — but of course, it should also be described there.

In English text, you write “Figure 3,” not “figure 3,” since it functions as a name for the figure (and similarly for Table, Section, Appendix, etc.).

All figures and images that are not your own must have references to the source.

If you include code snippets, make sure they are relevant and commented, so they can be understood. Alternatively, for short snippets:

provide the corresponding explanation in the text. In Typst you use `raw` elements for this purpose, see the code snippet in *Section B.5.2*. In this project supported languages are formatted using codly.

## B.4 Examples

Some examples for different commands in Typst.

### B.4.1 References

Uppsala University has 52 917 students *[3]*.

Bell et al. *[1, p. 312]* describes…

*Chapter B.4* outlines … and check out *Appendix B* as well as *Section B.5*.

### B.4.2 Acronyms

**NOTE:** Example usage of Acronyms. Acronyms list can be seen in abbreviations section as it can't be displayed twice.

**Constrained Equations**

Boundary Conditions (**BCs**) constrain the Degrees of Freedom (**DOFs**) of the Partial Differential Equations (**PDEs**) they act on.

BCs constrain the DOFs of the PDEs they act on.

The Method of Lines↔(MOL) is a procedure to solve PDEs in time.

**A note about trains showcasing alt**

The High speed train↔(TGV, from French *Train à haute vitesse*) has a commercial speed of up to 320 km/h.

## B.5 Figures

Look at this fabulous building and statue in *Figure 1*.

**Figure 1** University Hall or the University Main Building is the main building of Uppsala University in Uppsala, Sweden. The statue in front of the building is Erik Gustaf Geijer (1783-1847), a Swedish writer, historian, poet, philosopher, and composer. Photo: ScanianDragon (2023).

### B.5.1 Tables

Tables can also be placed in figures as seen in *Table 1*.

**Table 1** Number of students and employees at Uppsala University.

| | **Students** | **Employees** |
|---|---|---|
| *Quantity* | 52 917 | 7 622 |

Alternativly use the `tblr` package for more complex tables, such as the booktabs style seen in *Table 2* from the project page that also includes more info.

**Table 2** This is a caption

| | tol $= \mu_{\text{single}}$ | | | tol $= \mu_{\text{double}}$ | | |
|---|---|---|---|---|---|---|
| | $mv$[a] | Rel. err | Time[b] | $mv$[a] | Rel. err | Time[b] |
| *trigmv*[†] | 11034 | 1.3e-7 | 3.9 | 15846 | 2.7e-11 | 5.6 |
| *trig-expmv* | 21952 | 1.3e-7 | 6.2 | 31516 | 2.7e-11 | 8.8 |
| *trigblock* | 15883 | 5.2e-8 | 7.1 | 32023 | 1.1e-11 | 1.4e1 |
| *expleja* | 11180 | 8.0e-9 | 4.3 | 17348 | 1.5e-11 | 6.6 |

[a] $mv$ is in kg·m².
[b] Time is in secs.
[†] Another note.
*Note:* Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam.

### B.5.2 Code

Finally, you can insert code in a figure as in *Listing 1*. I like `codly` for syntax highlighting.

```python
# Sum
def sum(x, y):
    return x + y

print("Sum:", sum(1,1))
```

**Listing 1** Python implementation of adding two numbers.

### B.5.3 Math

You can decide if you want math in text such as $a^2 + b^2 = c^2$ or as a block like in *Equation 1*.

$$A = \frac{\pi r^2}{2} = \frac{1}{2}\pi r^2 \tag{1}$$

Placing whitespace after the opening dollar sign and before the closing dollar sign is the shorthand for blocking. You can also add alt text using the `#math.equation(alt: "alt text", content)` syntax. If you want the equation syntax to block use the block option `#math.equation(block: true)` as in *Equation 2*

$$\int_1^\infty ax^2 + b \, \mathrm{d}x \tag{2}$$

**Example of both in text match and blocks**

Let $a$, $b$, and $c$ be the side lengths of right-angled triangle. Then, we know that:

$$a^2 + b^2 = c^2 \tag{3}$$

Prove by induction:

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2} \tag{4}$$

### B.5.4 Timelines in typst

It is always nice to have timelines in reports. For this purpose I prefer using the `timeliney` package. Here is an example of a timeline for a optimization project and more info can be found on the project page: