le cnam

Strucutral dynamics

Template for the literature review report

Author
Mathieu Aucejo

Master in Structural Mechanics and Coupled Systems

Academic year 2024 – 2025

Table of contents

Abstract

This Typst package is a template for writing a literature review report as part of the "Structural Dynamics" module of the master's program "Structural Mechanics and Coupled Systems" at the Conservatoire national des arts et métiers.

1. What is Typst ?

Typst is a new open source markup language written in Rust and developed by two German students, Laurenz Mädje and Martin Haug, as part of their master's project [1], [2] from 2019. Version 0.1.0 was released on GitHub on April 04, 2023.

Typst presents itself as a more modern, faster and easier to use successor of LaTeX. Its advantages include

- incremental compilation;
- clear, understandable error messages;
- a Turing-complete programming language
- a coherent styling system that makes it easy to configure all aspects of your document (font, pagination, margins, ...);
- an active, friendly community (discord server for support, new package announcements);
- an easy-to-use package system (to search or view the list of packages, visit Type: Universe
- There are extensions for VS Code, such as `Typst LSP` or `Tinymist`, for features similar to `LaTeX Workshop`.

Finally, the Typst documentation is so well written and detailed that you can quickly create your own documents. It takes less than an hour to learn the syntax (no lie 😄). To access the documentation, follow this link. To ease the transition from LaTeX to Typst, a guide is available here.

2. Usage

To use the template, you must import it into your `typ` main file. Assuming the template and the main file are in the same folder, simply add the following command to the first line of the main file.

```
#import "report.typ": *
```

> If you split your document into multiple files, you must include the previous command as a preamble to each file.

2.1. Template initialization

After importing the model, it must be initialized by applying a `show` rule with the `#report()` command, passing the necessary options with the `with` instruction in your main `typ` file:

```
#show report-template.with(
  ...
)
```

The `#report-template()` template has a number of parameters to customize the document. Here is a list of available parameters:

```
#report-template(
  ⟨title⟩: [Titre du rapport],
  ⟨course⟩: [Nom du module],
  ⟨authors⟩: "",
  ⟨supervisors⟩: "",
  ⟨academic-program⟩: none,
  ⟨academic-year⟩: none,
  ⟨cover-image⟩: none,
  ⟨lang⟩: "fr"
)[⟨body⟩]
```

Argument

`⟨title⟩: [Titre du document]` — `str` | `content`

Title of the report

Argument

`⟨course⟩: [Nom du module]` — `str` | `content`

Name of the module

Argument

`⟨authors⟩: ""` — `array`

List of the authors

example : `("Author A", "Author B")`

Argument

`⟨supervisors⟩: ""` — `none` | `array`

List of the supervisors

example : `("Supervisor A", "Supervisor B")`

Argument

⟨academic-program⟩: none — none | str | content

Name of the academic program

Argument

⟨academic-year⟩: none — none | str | content

Current academic year

example : `"2024 - 2025"`

Argument

⟨cover image⟩: none — none | content

Cover image of the report

example : `image("image.png")`

2.2. Writing the document

The content of the document can be written in the main `typ` file or in auxiliary files. To learn the basics of writing a document in Typst, take a look at this tutorial, which will familiarize you with the main features in less than an hour.

2.3. Bibliography management

To insert a bibliography, insert the following command in the main `typ` file:

```
#bibliography("bliblio-file.yml/bib")
```

> The template provides two bibliography formats: YAML and BibTeX.
>
> The YAML file is interpreted by the `hayagriva` module, whose documentation is available here.

To cite a bibliographic reference in the text, simply use the `#cite(⟨key⟩)` command, or more simply `@key` (where `key` is the key associated with the reference).

For more information on managing bibliographic references, see the documentation for the `#bibliography()` function in Typst (accessible here).

2.4. Appendix management

The template provides an `appendix` environment to distinguish between the body of the document and the appendix information. In this environment the numbering of section headings, figures, tables and equations is adapted to the context. To enable this environment, add the following command to the main `typ` file at the desired location:

```
#show: appendix
```

2.5. Additional features

This section introduces the additional features implemented in the template to facilitate document editing.

Figures

Typically, figures are inserted into the document using Typst's `#figure()` function. However, Typst currently lacks mechanisms for managing sub-figures (numbering and referencing). To overcome this shortcoming, the template defines a `#subfigure()` function to manage subfigures in an appropriate way. This function encapsulates the `#subpar.grid()` function in the `subpar` package.

```
#subfigure(
  figure(image("image1.png"), caption: []),
  figure(image("image2.png"), caption: []), <b>,
  columns: (1fr, 1fr),
  caption: [Title of the figure],
  label: <fig:subfig>,
)
```

The previous example illustrates the case of a figure that consists of two sub-figures. The first has a caption, while the second has a `label` but no caption. The second sub-picture can therefore be referenced in the text with the command `@b`.

Equations

To box an important equation, use the `#boxeq()` function.

```
$
#boxeq[$p(A|B) prop p(B|A) space p(A)$]
$
```

To create an equation without numbering, use the `#nonumeq()` function.

```
#nonumeq[$ integral_0^1 f(x) dif x = F(1) - F(0) $]
```

3. Recommended packages

This section presents a list of packages that may be relevant when writing a document in Typst.

Drawing – `CeTZ`
- **Description**: This package is intended to be a Typst equivalent of TiKZ for LaTeX.
- **Links** - Typst: Universe, GitHub repository and documentation.

Boxes – `showybox`.
- **Description**: This package creates customizable content boxes (text, ...).
- **Links** - Type: Universe, GitHub repository and documentation.

Code – `codelst`
- **Description** : This package allows you to format blocks of source code, including line numbering.
- **Links** Typst: Universe, GitHub repository and documentation.

Algorithm – `lovelace`
- **Description**: This package allows you to write pseudocode with customizable formatting.
- **Links** - Type: Universe, GitHub repository and documentation (see ReadMe on GitHub).

Mathematics – `physica`
- **Description**: This package provides shortcuts for writing mathematical symbols.
- **Links** - Type: Universe, GitHub repository and documentation.

Glossary - `glossarium`.
- **Description**: This package will create a glossary.
- **Links** - Type: Universe, GitHub repository and documentation.

Index - `in-dexter`.
- **Description**: This package makes it easy to create an index.
- **Links**: Type: Universe, GitHub repository and documentation.

Presentation - `polylux`
- **Description**: This package allows you to create `PowerPoint` or `Beamer`-like presentations.
- **Links** : Type: Universe, GitHub repository and documentation.

Bibliography

[1] L. Mädje, "A Programmable Markup Language for Typesetting," M.S. Thesis, Technische Universität Berlin, 2022.

[2] M. Haug, "Fast Typesetting with Incremental Compilation," M.S. Thesis, Technische Universität Berlin, 2022.