東京大学大学院 工学系研究科 航空宇宙工学専攻

修士論文

Typst で書く修論のテンプレ

12-345678 右往 左往

指導教員: 魚 竿 准教授

2025 年 11 月 26 日 提出

概 要

近年の宇宙ってほんますごい. 近年の宇宙ってほんますごい. 近年の宇宙ってほんますごい. 近年の宇宙ってほんますごい. 近年の宇宙ってほんますごい. 近年の宇宙ってほんますごい. 近年の宇宙ってほんますごい. 近年の宇宙ってほんますごい. 近年の宇宙ってほんますごい. 近年の宇宙ってほんますごい. 近年の宇宙ってほんますごい.

キーワード: 宇宙, 異常検知

目 次

図 目 次

表 目 次

第 1 章　序論

Typst [1] は、Markdown のような分かりやすい記法で、PDF 文書・ポスター・スライド等の各種ドキュメントを簡単に作成できます。Rust 言語で書かれており、LaTeX に比べてコンパイルが極めて高速なのが特長です.

1.1　Typst は優秀だ

1.1.1　簡単に書ける

`$ mat(1, 2; 3, 4) $ <eq1>` と書くと、式 (1.1) を書くことができます。`<eq1>` はラベル名で、コンパイルすると数式への文書内リンクに置換されます。

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \tag{1.1}$$

実にシンプルですね。`\begin{pmatrix}` みたいなことを書く必要はなく、Markdown + MathJax のように `$ $` だけで良いんです。

分数やカッコもお手の物。`\frac{a}{b}` なんてややこしい記法は使わず、ただ `$ a / b $` と書けば分数になります。`\left(` とか `\right)` とかを自分で書かなくても、Typst はカッコの対応関係を自動で検知し、良い感じにサイズを合わせてくれます。

```
$ F_n = 1 / sqrt(5) dot ( ( (1 + sqrt(5)) / 2) ^ n - ((1 - sqrt(5)) / 2) ^ n ) $
```

$$F_n = \frac{1}{\sqrt{5}} \cdot \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right) \tag{1.2}$$

```
$ f(x, y) := cases(
  1 "if" (x dot y)/2 <= 0,
  2 "if" x "is even",
  3 "if" x in NN,
  4 "else",
) $
```

$$f(x, y) := \begin{cases} 1 \text{ if } \frac{x \cdot y}{2} \leq 0 \\ 2 \text{ if } x \text{ is even} \\ 3 \text{ if } x \in \mathbb{N} \\ 4 \text{ else} \end{cases} \tag{1.3}$$

画像や表の挿入も簡単です。次のようにすると 図 1.1 を表示できます。

```
#img(
  image("Figures/typst.svg", width: 20%),
```

```
  caption: [Typstのロゴ],
) <img1>
```

typst

図 1.1 Typst のロゴ

表 1.1 はこんな感じ。

```
#tbl(table(
    columns: 4,
    [t], [1], [2], [3],
    [y], [0.3s], [0.4s], [0.8s],
  ),
  caption: [テーブル @madje2022programmable],
) <tbl1>
```

表 1.1 テーブル [1]

t	1	2	3
y	0.3s	0.4s	0.8s

こんな感じで [2] or [2] と引用できます。引用方式も数十種類の中から選べます。

```
こんな感じで @ss8843592 or #cite(<ss8843592>) と引用できます。
```

また、文中に簡単なプログラムを直接埋め込むことも可能です（コンパイル時に評価されて計算結果がテキストに変換される）。

他にも `#include path.typ` とすれば他ファイルを参照できます。テンプレートファイルを作って別のファイルから呼び出したり、長い分量の本などを作成する際に章ごとにファイルを分けることなどができます。

LaTeX は世界中のユーザによる膨大な資産と、長年かけて築いてきた圧倒的なシェアがあるため、すぐに Typst に取って代わることはないでしょう。しかし講義ノート・卒論/修論・学会の予稿等の作成などの場面では、少しずつ Typst に置き換わっていくでしょう（願望）。

t + 

図 1.2 Typst + git [1]

1.2　グラフィックスも色々できるよ

このように、Typst はデフォルトでも様々なグラフィックス機能を備えていますが、他にも「CetZ」というパッケージがあります（TikZ の Typst 版のようなもの）。これを使うと、もっと自由度の高い図を色々と描くことができます。

```
#import "@preview/cetz:0.4.2": canvas, draw, vector, matrix

// #set page(width: auto, height: auto, margin: .5cm)

#canvas({
  import draw: *

  ortho(y: -30deg, x: 30deg, {
    on-xz({
      grid((0,-2), (8,2), stroke: gray + .5pt)
    })

    // Draw a sine wave on the xy plane
    let wave(amplitude: 1, fill: none, phases: 2, scale: 8, samples: 100) = {
      line(..(for x in range(0, samples + 1) {
        let x = x / samples
        let p = (2 * phases * calc.pi) * x
        ((x * scale, calc.sin(p) * amplitude),)
      }), fill: fill)

      let subdivs = 8
      for phase in range(0, phases) {
        let x = phase / phases
        for div in range(1, subdivs + 1) {
          let p = 2 * calc.pi * (div / subdivs)
          let y = calc.sin(p) * amplitude
          let x = x * scale + div / subdivs * scale / phases
          line((x, 0), (x, y), stroke: rgb(0, 0, 0, 150) + .5pt)
        }
      }
    }

    on-xy({
      wave(amplitude: 1.6, fill: rgb(0, 0, 255, 50))
    })
    on-xz({
      wave(amplitude: 1, fill: rgb(255, 0, 0, 50))
    })
  })
})
```

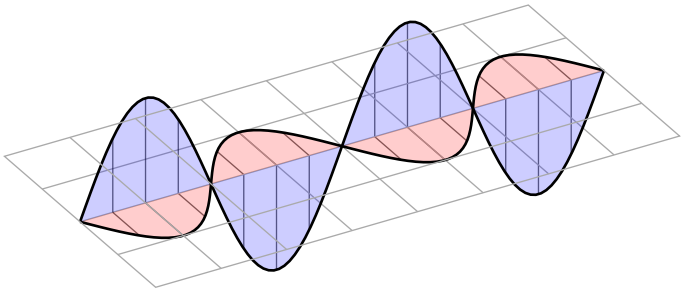

図 1.3 CetZ で描いた図の例（公式 GitHub リポジトリより引用）

第 2 章　自分でスタイルを定義する

Typst では定理の書き方などをカスタマイズできます.

2.1　実例

`thmbox` 関数を作ってカスタマイズをできるようにしました.

```
#let theorem = thmbox(
  "theorem", //identifier
  "定理",
  base_level: 1
)

#theorem("オイラー")[
  Typst はすごいのである.
] <theorem>
```

定理 2.1 (湯川). セガなんてダッセーよな！

```
#let lemma = thmbox(
  "theorem", //identifier
  "補題",
  base_level: 1,
)

#lemma[
  帰ってTypstやろーぜー！
] <lemma>
```

補題 2.2. 帰って Typst やろーぜー！

このように, 定理 2.1 , 補題 2.1 を定義できます.

カッコ内の引数に人名などを入れることができます.また, identifier を変えれば, カウントはリセットされます。 identifier 毎にカウントを柔軟に変えられるようにしてあるので, 様々な論文の形式に対応できるはずです.

```
#let definition = thmbox(
  "definition", //identifier
  "定義",
  base_level: 1,
  stroke: black + 1pt
)
```

```
#definition("Prime numbers")[
  A natural number is called a _prime number_ if it is greater than $1$ and
  cannot be written as the product of two smaller natural numbers.
] <definition>
```

定義 2.1: Typst is a new markup-based typesetting system for the sciences.

このように、「定義 2.1 」のカウントは「2.1」にリセットされていますね。

```
#let corollary = thmbox(
  "corollary",
  "Corollary",
  base: "theorem",
)

#corollary[
  If $n$ divides two consecutive natural numbers, then $n = 1$.
] <corollary>
```

Corollary 2.2.1: If n divides two consecutive natural numbers, then $n = 1$.

base に identifier を入れることで Corollary 2.1 のようにサブカウントを実現できます.

```
#let example = thmplain(
  "example",
  "Example"
).with(numbering: none)

#example[
  数式は\$\$で囲む
] <example>
```

例: 数式は $$ で囲む

thmplain 関数を使って plain 表現も可能です.

付録 A　こういう機能もいるよね

　コンテンツの周囲を `#appendix[]` で囲むと、そのコンテンツはそのまま付録セクションになります。ナンバリング方式もアルファベットに変わります。

```
#appendix[
  = こういう機能もいるよね

  コンテンツの周囲を `#appendix[]` で囲むと、そのコンテンツはそのまま付録セクションになります。ナンバリング方式もアルファベットに変えてありますよ！
]
```

参考文献

[1] L. Mädje, “A Programmable Markup Language for Typesetting,” Doctoral dissertation, 2022.

[2] S. Hussain, S. Bai, and S. Khoja, “Content MathML(CMML) conversion using LATEX Math Grammar (LMG),” in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, 2019, pp. 1–5. doi: 10.1109/ICSCC.2019.8843592.