

Typst を使った国内学会論文の書き方

- 国内学会予稿集に似せたフォーマットの作成 -

○ 著者姓 1 著者名 1, 著者姓 2 著者名 2(○○○大学), 著者姓 3 著者名 3 (□□□株式会社)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1. はじめに

これは非公式のサンプルです。 適宜投稿先の規定をご確認ください。 発表論文原稿を PDF でご執筆いただき、学会のホームページにアップロードしてください。 このファイルはこのテンプレートの使い方を示しており、同時に発表論文の見本でもあります。 執筆の時は以下の説明をよく読み、執筆要項に従ったフォーマットでご提出ください。 アップロードした PDF がそのまま公開されます。 というような説明が書かれるであろうテンプレートを作ってみました。 本稿では、このテンプレートファイルの使い方および Typst による執筆作業の概要について解説します。 この原稿のソースコードは <https://github.com/kimushun1101/typst-jp-conf-template> で公開しております。 Typst の概要についてお知りになりたい方は、<https://github.com/kimushun1101/How-to-use-typst-for-paper-jp> にもスライド形式の資料を用意しておりますので、ぜひこちらもご覧ください。

2. テンプレートファイルの使い方

GitHub に慣れていればテンプレートリポジトリを使用して、新しいリポジトリを作成してクローン。 不慣れであれば zip ダウンロードして展開してください。

テンプレートファイルは以下の 2 つの方法で実行できることを確認しています。

1. VS Code とその拡張機能を使う
2. 好みのエディターで編集した後 CLI (Command Line Interface) でビルドする

2.1. Visual Studio Code による執筆

コマンドライン入力に忌避感のある方は（またそうでない方も） Visual Studio Code (VS Code) の使用をオススメします。 VS Code の拡張機能である Tinymist Typst をインストールすれば、編集中でも現在の出力結果を常に確認することができます。 また、`.vscode/settings.json` にて保存と同時に PDF ファイルが作成される設定にしております。

2.2. Typst CLI によるビルド

2.2.1. インストール

- Windows の場合
Windows PowerShell から以下のコマンドでインストールできる。

```
1 winget install --id Typst.Typst
```

- Mac の場合
Homebrew を使ってインストールできる。

```
1 # Homebrew のインストール
2 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
3 # Typst のインストール
4 brew install typst
```

- Rust からインストール
たとえば Ubuntu の場合は、Rust の cargo を使ってインストールする方法が簡単と思われ
ます。

```
1 # Rust のインストール
2 curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
3 # Typst のインストール
4 cargo install --git https://github.com/typst/typst --locked typst-cli
```

2.2.2. ビルド

シェルで対象のディレクトリに移り

```
1 typst compile main.typ
```

とコマンドすれば main.pdf をビルドできます。

2.3. このパッケージが使えない場合

ビルドができない場合などのトラブルがございましたら、<https://github.com/kimushun1101/typst-jp-conf-template/issues> でご相談ください。 以下に対応方法を記載します。

2.3.1. パッケージの自動インストールができない場合

大学や会社などのプロキシ環境によっては、パッケージの自動インストールがブロックされてしまう

可能性がある。その場合には Tyspt Universe から圧縮ファイルを手動でダウンロードする。本サンプルで使用しているパッケージを入手できるリンクは以下のとおりである。

- <https://typst.app/universe/package/sourcerer>
- <https://typst.app/universe/package/ctheorems>

これらの圧縮ファイルを main.typ と同じフォルダーにある libs フォルダの中に展開した後、以下のようにコメントアウトを付け替えて、それぞれの lib.typ ファイルへのパスを指定する。

```
1 // ソースコードブロックを表示するためのパッケージ
2 // #import "@preview/sourcerer:0.2.1":
  code
3 #import "libs/sourcerer-0.2.1/src/
  lib.typ": code // 2.3.1 を参照
4
5 // 定理環境
6 // #import "@preview/ctheorems:1.1.2":
  thmplain, thmproof, thmrules
7 #import "libs/ctheorems-1.1.2/lib.typ":
  thmplain, thmproof, thmrules // 2.3.1 を参照
```

3. 原稿の体裁

3.1. レイアウトとフォント

用紙サイズは A4, 縦 250 mm, 横 170 mm の枠内に収まるようにしてください。余白は、上 20 mm, 下 27 mm, 左 20 mm, 右 20 mm とします。タイトル, 著者, アブストラクトはシングルコラム, 本文はダブルコラムです。アブストラクトは左右に 0.7 cm 余白を取っています。

フォントの設定は表 1 の通りです。ここで、ゴシック体とは“BIZ UDPGothic”, “MS PGothic”, “Hiragino Kaku Gothic Pro”, “IPAex-Gothic”, “Noto Sans CJK JP” のいずれか, 明朝体とは“BIZ UDPMincho”, “MS PMincho”, “Hiragino Mincho Pro”, “IPAexMincho”, “Noto Serif CJK JP” のいずれかで見つかるものが採用されます。これらのフォントがお使いのコンピュータになればインストールするか, 代わりに使いたいフォントがあればソースコードの方に追加してください。以下のコマンドで使用可能なフォント一覧を確認できます。

```
1 typst fonts
```

今回使用しているテンプレートは“rsj-conf/lib.typ”に記載されています。このテンプレートのソースコードは charge-ieee と abiding-ifacconf というテンプレートを参考にして、第 41 回日本ロボット学会学術講演会のフォーマットに近づけて作成しました。テンプレートの検索

は Typst Universe <https://typst.app/universe> ででき、そこで掲載されているテンプレートのソースコードは <https://github.com/typst/packages/tree/main/packages/preview> で見るができます。

3.2. 数式・図・表

数式番号は (1) のように数式の右側に、図のタイトルは“図 1 タイトル名”のように図の下部に、表のタイトルは“表 1 タイトル名”のように図の上部につきます。投稿先に応じてキャプションの言語は日本語や英語で指定されるかと思しますので、指示に従ってください。

3.3. 定理環境

以下は、theorem 環境の使用例です。現バージョンでは日本語に太字が使えない代わりに、フォントをゴシックにすることでそれっぽく見せています。

定義 1 (用語 A) 用語 A の定義を書きます。

補題 1 補題を書きます。タイトルは省略することもできます。

補題 2 (補題 C) 補題を書きます。番号は定義や補題ごとに 1 からカウントします。

定理 1 (定理 D) ここに定理を書きます。

系 1 系を書きます。定義 1 のように、ラベルで参照することもできます。

証明 定理 1 の証明 証明を書きます。証明終了として□印をつけています。□

3.4. 特殊な章

謝辞と参考文献は他の章とは異なり、章番号が自動でつかないように設定しています。また、参考文献は“参考文献”とスペースで区切り、中央揃えにしています。

4. 編集の仕方

4.1. 論文情報の編集

main.typ の文頭にある以下のコードを解説します。

表 1 フォントの設定

項目	サイズ (pt)	フォント
タイトル	18	ゴシック体
著者名	12	ゴシック体
章タイトル	12	ゴシック体
節, 小節, 本文	10	明朝体
参考文献	9	明朝体

```

1 #import "libs/rsj-conf/lib.typ": rsj-
  conf
2 #show: rsj-conf.with(
3   title: [Typst を使った国内学会論文の書き方 \
  - 国内学会予稿集に似せたフォーマットの作成 - ],
4   authors: [○ 著者姓1 著者名1, 著者姓2 著者
  名2(○○大学), 著者姓3 著者名3 (○○株式会社)],
5   abstract: [#lorem(80)],
6   bibliography: bibliography("refs.yml",
  full: false)
7 )

```

1 行目はこの原稿の体裁を設定するためのソースコードを import しています。これは“libs”ディレクトリ以下にあります。2 行目は、ソースコードやコマンドなどを綺麗に表示するための“code”関数を呼び出すために import しています。こちらは“libs”ディレクトリにはありません。Typst Universe から自動でインストールされたものを使っております。その他、Typst Universe で使いたい packages があるところで同様に import しておくといよいでしょう。4 行目では、1 行目で読み込んだ関数を使用して、原稿体裁を作成しています。5 行目ではタイトルを、6 行目では著者一覧を、7 行目ではアブストラクトを記載します。各内容の中で改行をしたい場合には、\ で改行してください。lorem 関数は英文のダミーテキストを作成してくれる関数です。8 行目の参考文献については本章の最後の節で説明します。

また、異なるテンプレートも用意してみました。コメントアウトで切り替えてみてください。

```

1 #import "libs/rengo/lib.typ": rengo
2 #show: rengo.with(
3   title: [Typst を使った国内学会論文の書き方
  \ - 国内学会予稿集に似せたフォーマットの作成 - ],
4   authors: [○ 著者姓1 著者名1, 著者姓2 著者
  名2(○○大学), 著者姓3 著者名3 (○○株式会社)],
5   etitle: [How to write a conference
  paper in Japanese],
6   eauthors: [\*A. First, B. Second (○○
  Univ.), and C. Third (○○ Corp.)],
7   abstract: [#lorem(80)],
8   keywords: ([Typst], [conference paper
  writing], [manuscript format]),
9   bibliography: bibliography("refs.yml",
  full: false)
10 )

```

このフォーマットですと、etitle, eauthors, keywords が追加されており、それぞれ英語タイトル、英語著者名、キーワードを意味しています。keywords は () のリスト形式で指定されていることに注意してください。

#import でテンプレートの関数を持ってくるところと、その関数を使用するところ以外の本文部分のコードはテンプレートの変更に応じて変更する必要はありません。

4.2. 基本的な文法

章は =, 節は ==, 小節は === で始めます。改段落は LaTeX と同じく改行を 2 つ以上挟みます。数字つき箇条書きは + で、数字なしの箇条書きは - を文頭につけて書くことができます。テキストの装飾は text 関数で行えます。LaTeX に慣れている方は、Typst 公式ページの <https://typst.app/docs/guides/guide-for-latex-users/> を読むと雰囲気がつかめると思います。

4.3. 数式

数式番号をつけるような中央揃えの数式は、最初の \$ の後ろと閉じの \$ の前にスペースを挟み

```

1 $ dot(x) &= A x + B u \
2 y &= C x $ <eq:system>

```

のように書き、文中に書く数式は、\$ の前後にスペースを挟まず

```

1 $x in RR^n$

```

というように書きます。ここで <eq:system> は引用するときのラベルになります。

出力例はつぎの通りです。以下のシステムを考える。

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (1)$$

ここで $x \in \mathbb{R}^n$ は状態, $u \in \mathbb{R}^m$ は入力, $y \in \mathbb{R}^l$ は出力, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$. および $C \in \mathbb{R}^{l \times n}$ は定数行列である。このシステムに対して、目標値 $r(t)$ に対する偏差を $e = r - y$ とした以下の PI 制御器を使用する。

$$u = K_P e + K_I \int_0^t e dt \quad (2)$$

ただし、 K_P と K_I はそれぞれ比例ゲイン、積分ゲインとする。

4.4. 図と表

本稿を執筆時のバージョン Typst 0.11.0 では、PNG, JPEG, GIF, SVG の形式のイメージデータを挿入することができます。例としては以下の通りです。

```

1 #figure(
2   placement: bottom,
3   image("figs/quadratic.svg", width:
  90%),
4   caption: [$x^2$ のグラフ],
5   ) <fig:quadratic>
6
7 #figure(
8   placement: bottom,
9   image("figs/sqrt-and-sin.png", width:
  90%),
10  caption: [$\sqrt{x}$ と $\sin x$ のグラフ],
11  ) <fig:sqrt-sin>

```

ここで placement は、紙面の上 (top) に寄せるか下 (bottom) に寄せるかを決められます。言及している文章に近い方に調整してください。

表 1 は以下で記述されております。

```

1 #figure(
2   placement: top,
3   caption: [フォントの設定],
4   table(
5     columns: 3,
6     stroke: none,
7     table.header(
8       [項目],
9       [サイズ (pt)],
10      [フォント],
11    ),
12    table.hline(),
13    [#text(18pt, "タイトル", font:
14      gothic)], [18], [ゴシック体],
15    [#text(12pt, "著者名", font:
16      gothic)], [12], [ゴシック体],
17    [#text(12pt, "章タイトル")], [12], [ゴ
18      シック体],
19    [節, 小節, 本文], [10], [明朝体],
20    [#text(9pt, "参考文献")], [9], [明朝
21      体],
22  )
23 ) <tab:fonts>

```

table の columns の数に応じて、文字列の配列が自動的に整列されます。stroke: none は枠線を消しています。table.hline() を挟むとその位置に横線を引けます。ここで, gothic は lib.typ で定義されています。他のテンプレートを使用する場合には注意をしてください。

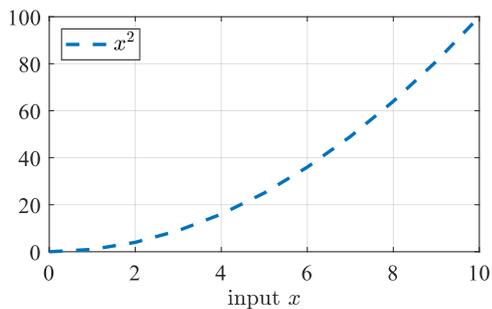


図 1 x^2 のグラフ

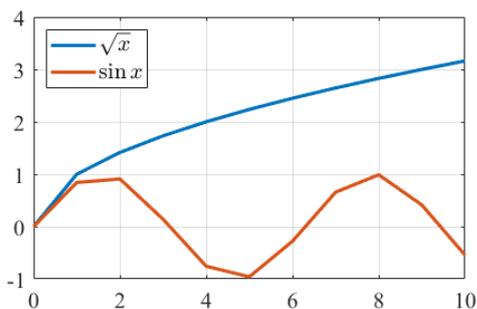


図 2 \sqrt{x} と $\sin x$ のグラフ

```

1 #import "libs/rsj-conf/lib.typ": rsj-
2   conf, gothic

```

4.5. 定理環境

定義 1 や 補題 1 などは以下で記述されております。

```

1 #definition("用語 A")[
2   用語 A の定義を書きます。
3 ]<def:definition1>
4 #lemma[
5   補題を書きます。タイトルは省略することもでき
6   ます。
7 ]<lem:lemma1>
8 #lemma("補題 C")[
9   補題を書きます。番号は定義や補題ごとに 1 から
10  カウントします。
11 ]<lem:lemma2>
12 #theorem("定理 D")[
13   ここに定理を書きます。
14 ]<thm:theorem1>
15 #corollary[
16   系を書きます。@def:definition1 のように、ラ
17   ベルで参照することもできます。
18 ]
19 #proof([@thm:theorem1 の証明])[
20   証明を書きます。証明終了として□印をつけていま
21   ます。
22 ]

```

ここで, definition, lemma, theorem, corollary, proof は gothic と同様に lib.typ で定義されています。他のテンプレートを使用する場合には注意をしてください。

```

1 #import "libs/rsj-conf/lib.typ": rsj-
2   conf, gothic

```

さらに元をたどると lib.typ で ctheorems パッケージ (<https://typst.app/universe/package/ctheorems>) をインポートして使用しております。

```

1 // Theorem environment
2 #import "@preview/ctheorems:1.1.2":
3   thmplain, thmproof, thmrules
4 #let thmjp = thmplain.with(base: {},
5   separator: [#h(0.5em)], titlefmt: strong,
6   inset: (top: 0em, left: 0em))
7 #let definition = thmjp("definition",
8   text(font: gothic)[定義])
9 #let lemma = thmjp("lemma", text(font:
10   gothic)[補題])
11 #let theorem = thmjp("theorem",
12   text(font: gothic)[定理])
13 #let corollary =
14   thmjp("corollary", text(font: gothic)[系])
15 #let proof = thmproof("proof", text(font:
16   gothic)[証明], separator: [#h(0.9em)],
17   titlefmt: strong, inset: (top: 0em, left:
18   0em))

```

4.6. 参考文献

参考文献は refs.yml に記載してください。Haya-griva という YAML 形式のフォーマットに従っています。編集するだけであれば特に解説する必要はないと思います。詳細が気になる方は <https://github.com/typst/hayagriva> をご参照ください。参考文献の体裁は libs/rsj-conf/bib.csl で制御しています。これは Citation Style Language という XML 形式で記述されております。CSL ファイルは著者が編集する必要はありませんが、詳細が気になる方は <https://citationstyles.org/> をご参照ください。日本語論文として重要な点は、CSL ファイルでは Hayagriva で記述された language の属性を見て、著者表示を"カンマ区切りのみ"にするか"カンマ区切り+最終著者の前に and" にするかを決定している点です。したがって、英語文献だけでしたら YAML ファイルの代わりに bib ファイルも使用することができます。

文献内で引用された順番にフォーマットを整えて自動で参考文献の章が作られます。引用の方法については後述します。完成原稿では推奨されませんが、引用されていない論文も記載したい場合には full: true にすれば、すべての文献が出力されます。

4.7. 引用

引用は "@label" と記述することで、数式であれば (1)、図であれば 図 1、表であれば 表 1、参考文献であれば [1] のように表示されます。参考文献は連続して引用すると [2]–[7] と繋げられて表示されます。文法上では特に規則はありませんが、個人的にはラベルの命名規則として、図の場合には "fig:" から、表の場合には "tab:" から始めるようにラベル名を設定しており、参考文献のラベルは "著者名発行年タイトルの最初の単語" で名付けております。

5. おわりに

筆者の理解や表現が誤っている箇所もあるかと思えます。GitHub を通して、Issues や Pull Requests を歓迎しております。日本語での投稿で構いません。誤字脱字や文法、表現など細かい修正でも大変ありがたいです。筆者は、Typst が普及するためには学会のフォーマットで配布されることが不可欠だと感じています。異なる学会のフォーマットも随時 libs ディレクトリに追加していこうと思っております。これらのファイルが Typst が普及の一助となれば幸いです。

謝辞

謝辞には章番号が振られないように設定しております。「この研究は☆☆☆の助成を受けて行われました。」や「〇〇〇大学との共同研究です。」みたいな文章が書かれることを想定しています。最後までお読みいただき誠にありがとうございました。

参 考 文 献

- [1] S. Kimura, H. Nakamura, and Y. Yamashita, "Asymptotic stabilization of two-wheeled mobile robot via locally semiconcave generalized homogeneous control Lyapunov function," *SICE Journal of Control, Measurement, and System Integration*, vol. 5, no. 2, pp.122–130, 2015
- [2] 木村, 中村, 伊吹, 三平, "状態制約を有する非線形システムに対する状態方程式を保持する変換—システム蘇生変換—," 計測自動制御学会論文集, vol. 53, no. 6, pp.337–345, 2017
- [3] S. Kimura and K. Naito, "Control barrier function based assist control with shifting sensor and footprint positions," *IECON2021 - 47th annual conference of the IEEE industrial electronics society*, 2021
- [4] 木村, "施設管理システムと自律移動ロボットとの連携実証試験," ロボティクス・メカトロニクス講演会講演概要集, 2020
- [5] H. K. Khalil, *Control of nonlinear systems*, Global Edition. Pearson Education, 2002
- [6] 杉江, 藤田, フィードバック制御入門. コロナ社, 1999
- [7] "視覚障がい者移動支援ロボット「AIスーツケース」," 清水建設株式会社, Accessed on 30.03.2024 URL: <https://www.shimz.co.jp/solution/tech374/index.html>.