

20\*\*年度修士論文

【非公式】機械航空宇宙工学科 学位論文テンプレート  
—LaTeXで論文を書く際に必要な最低限の情報—

20\*\*年2月

東京理科大学大学院創域理工学研究科機械航空宇宙工学専攻

○○研究室

75\*\*\*\* 姓姓 名名

# 目次

記号表	.....	i
第1章	はじめに	1
1.1	テンプレート概要	1
1.2	リポジトリ内のファイル構成	3
1.3	卒論・修論要旨	4
第2章	環境構築・操作方法	5
2.1	環境構築	5
2.1.1	Windows の場合	5
2.1.2	macOS の場合	6
2.2	使用するエディター	7
2.3	pdf ファイルの生成	7
2.3.1	ターミナル上での操作	7
2.3.2	pdflATEX で日本語を使用する場合	10
2.3.3	latexmk を使う方法	11
2.3.4	クラウド上で使う方法	11
第3章	LATEX の基本	13
3.1	LATEX とは何か	13
第4章	図表の配置	14
4.1	図の配置	14
4.1.1	図を 1 枚だけ配置する方法	14
4.1.2	図を複数枚配置する方法	15
4.1.3	画像のファイル形式	19
4.2	表の配置	21
4.2.1	tabular 環境を用いた表	22
4.2.2	tblr 環境を用いた表	23
第5章	BIBTEX による参考文献一覧の出力	25
5.1	参考文献の記載時の一般的な注意事項	25
5.1.1	引用方式	25

5.1.2	文献リストの作り方	26
5.1.3	文献の正しい引用の仕方	26
5.2	BIBTEX の使い方	26
5.2.1	jsme.bstについて	26
5.2.2	bib ファイルについて	27
第 6 章	先生や先輩に添削してもらうときの注意点	28
6.1	latexdiff	28
6.2	latexdiff-vc	28
第 7 章	さらに詳しい情報が欲しい人は	29
7.1	論文の書き方に関する情報	29
7.2	TeX/LaTeX に関する情報	29
7.2.1	書籍	29
7.2.2	インターネット上の情報	29
謝辞		30
文献		33
付録 A	修士課程における研究成果	34
付録 B	スーパーコンピューターごとの性能比較	36

# 記号表

## Alphabet

$d$	Channel width [m]
$L_j$	Computational domain size in $j$ -direction [m]
$N_j$	Number of grid points in $j$ -direction
$Re$	Reynolds number, $= ud/\nu$
$u$	Velocity [m/s]

## Greek

$\delta$	Channel half width [m]
$\epsilon_{ijk}$	Levi–Civita symbol
$\nu$	Kinematic viscosity [ $\text{m}^2/\text{s}$ ]

## Superscripts

$(\ )^*$	Normalized by outer variables, e.g., $\delta$
$(\ )^+$	Normalized by inner variables, e.g., $\nu/u_\tau$ (wall unit)
$(\ )'$	Fluctuation component
$\overline{(\ )}$	Statistically averaged

## Subscripts

$(\ )_{\text{rms}}$	Root mean square
$(\ )_{\text{w}}$	Wall
$(\ )_\tau$	Wall unit

# 第 1 章

## はじめに

第 1 章では学位論文執筆の際の注意事項として、第 1.1 節でこのテンプレートの概要を、第 1.2 節では GitHub リポジトリ内の各ファイルの説明を、第 1.3 節では卒論・修論要旨の L<sup>A</sup>T<sub>E</sub>X テンプレートの紹介をします。このテンプレートを使用する方は現在の L<sup>A</sup>T<sub>E</sub>X 習熟度によらず必ず目を通してください。

### 1.1 テンプレート概要

このファイルは東京理科大学創域理工学部機械航空宇宙工学科の卒業論文および同大学大学院創域理工学研究科機械航空宇宙工学専攻の修士論文を作成するにあたり、学科の論文執筆要件を満たした「非公式の」L<sup>A</sup>T<sub>E</sub>X テンプレートです。一連のファイルは東京理科大学創域理工学部機械航空宇宙工学科塙原研究室<sup>1)</sup>の GitHub Organization<sup>2)</sup> から入手可能です。塙原研究室は熱流体系の研究室ですが、所属研究室によらずこのテンプレートは使用可能です。パブリックリポジトリなので、他研究室所属の方もご自身の PC に入れることができます。また、使用する際に塙原研究室の許可を取る必要はありません。ご自由にお使いください。

このテンプレートは研究室に配属されて初めて L<sup>A</sup>T<sub>E</sub>X で文書を書くことになった学部 4 年生を対象に、環境構築から pdf ファイルの生成、卒業論文執筆まで滞りなく行えるように作成しています。そのため基本事項から説明をしていますが、表紙のタイトルにもある通り「必要最低限の情報」しか記載していません。L<sup>A</sup>T<sub>E</sub>X 入門書は既に良書がたくさんありますが、本当の初心者は知らなくてもいい情報や学位論文執筆だけを目指すうえでは不要な情報がたくさん書かれているため、困惑した読者も多いのではないかと思います。このテンプレートには学位論文執筆をするうえで学生が欲しがるであろう情報のみを厳選し、その情報とこのテンプレートだけあれば学位論文を書き上げるくらいのことはできるようにしておきました。そのため、T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X で一から文書を作成することを目指している方には情報が足りていないと思います。さらに詳しい情報が欲しい人は書籍やインターネット上の情報を参考にしてください（第 7 章を参照）。また、この main.pdf はモダンな L<sup>A</sup>T<sub>E</sub>X である L<sup>A</sup>u<sub>L</sub>T<sub>E</sub>X で作成しているほか、 j1req というドキュメントクラスや unicode-math など最新の機能をふんだん

---

1) 塙原研究室ウェブページ、<<https://www.rs.tus.ac.jp/~t2lab/index-j.html>>

2) TUS-ME\_thesis\_template、<[https://github.com/tsukahara-lab/TUS-ME\\_thesis\\_template](https://github.com/tsukahara-lab/TUS-ME_thesis_template)>

に使用しています。これからこのテンプレートを使い始めるという方はモダン  $\text{\LaTeX}$  を使えるようになっておきましょう。しかし、学会の講演論文執筆の際はこれらの機能に対応していない場合もあるため、念のためレガシーな  $\text{\LaTeX}$  のコンパイル方法等についても説明をしています。さらに、このテンプレートでは  $\text{\LaTeX}$  に関する説明はもちろんのこと、学生が論文を書くうえで躊躇やすい箇所をまとめています。特に表記に関して細かく記載しているので参考になる箇所は多いかと思います。

もしこのテンプレートに関してバグ等、使用上の問題が発生した際は GitHub の Issues にコメントしてください。ただし、このテンプレートを使用したことで生じた問題に関して大学・学科・塚原研究室および研究室に所属する個人は一切の責任を負いませんのでご了承ください。また、この文書に書かれている  $\text{\TeX}/\text{\LaTeX}$  技術に関する内容はできるだけ正確な記述を心掛けていますが、完全な正確性を保証するものではありません。

このテンプレートを使用される皆様が無事に学位論文を執筆し、卒業・修了されることを心の底から願っております。

2024年10月25日

塚原研究室 学生有志

## 1.2 リポジトリ内のファイル構成

tsukahara-lab/TUS-ME_thesis_template	
chapter/	分割した tex ファイルが入っているフォルダ
figure/	図が入っているフォルダ
table/	表の tex ファイルが入っているフォルダ
.gitignore	Git で管理しないファイル一覧
README.md	GitHub リポジトリの説明書
jsme.bst	日本機械学会対応の BIBTeX スタイルファイル
latexmkrc	詳細は第 2.3.3 節を参照
main.pdf	main.tex をコンパイルした pdf ファイル
main.tex	メインの文書ファイル
mybib_en.bib	英語の参考文献リストファイル
mybib_jp.bib	日本語の参考文献リストファイル
settings.sty	main.tex で読み込むスタイルファイル

README.md はこの GitHub リポジトリを開いたときに一番最初に目に入ってくる説明書です。内容をよく読んで使用するようにしてください。

今皆さんが出でているこの pdf ファイルは main.pdf で、 main.tex を基に作成しています。文書のレイアウト等細かい設定は全てスタイルファイル settings.sty に書いています。 main.tex 冒頭の \usepackage{settings} で読み込んでいるので間違って消さないようにしてください。 main.tex を適切なテキストエディター（第 2.2 節を参照）で開いてもらうと、 \include{chapter/xxx} と書かれた文字列が複数目に入ってくると思います。学位論文のような長い文書を一つの tex ファイルに書き込むとわかりにくくなるので、 chapter/ 以下のディレクトリに章 (chapter) ごとに分割した tex ファイルを置いておき、それを \include{} コマンドで読み込むようにしています。皆さんが出で論文を執筆する際にもこのように tex ファイルを分割しておきましょう。また、コンパイルの際には latexmk という機能を使用し、その際 latexmkrc が必要になります。 latexmk でコンパイルした際のファイル出力先を latex.out/ に設定しています。 latexmk の使用方法も含め、具体的なコンパイルの方法等については第 2 章を参照してください。

jsme.bst, mybib\_en.bib, mybib\_jp.bib は参考文献の出力に使用するファイル群です。具体的な使用方法は第 5 章を参照してください。

最後に、`.gitignore` は Git で管理しないファイルが書かれています。Git の詳細はここでは割愛しますが、`LATEX` で学位論文を執筆する際は Git でバージョン管理するようにしましょう。先生や先輩に添削してもらうときに前回見せたときとの差分を `latexdiff-vc`（第 6.2 節を参照）で見せることができるほか、GitHub のプライベートリポジトリに上げることでそれ自体がバックアップとなり、大変便利です。このリポジトリで使用している `.gitignore` は GitHub で `TeX/LATEX` に対して与えられる標準の `.gitignore` を使用しています。

### 1.3 卒論・修論要旨

卒業論文・修士論文を提出する際は同時に要旨が必要です。要旨についても `LATEX` テンプレートを作成したので、GitHub リポジトリ<sup>3)</sup>からダウンロードしてください。コンパイル方法はこのテンプレートと同様です。要旨に関する詳細な説明はここでは省略しますが、`README.md` にしっかりと目を通すようにしてください。

---

3) TUS-ME\_thesis\_abstract: <[https://github.com/Yuki-MATSUKAWA/TUS-ME\\_thesis\\_abstract](https://github.com/Yuki-MATSUKAWA/TUS-ME_thesis_abstract)>

# 第 2 章

## 環境構築・操作方法

第 2 章では  $\text{\TeX}/\text{\LaTeX}$  環境構築の方法と pdf ファイルの生成までのプロセスを説明します。第 2.1 節では TeX Live のインストール方法について、第 2.2 節では  $\text{\TeX}/\text{\LaTeX}$  対応のテキストエディター、特に VS Code の場合について述べ、第 2.3 節では pdf ファイル生成までに必要なコマンドや `latexmk` の使い方、クラウド上での  $\text{\LaTeX}$  の使用について述べます。

### 2.1 環境構築

$\text{\TeX}/\text{\LaTeX}$  を使用する際は TeX Live というディストリビューションをご自身の PC に入れましょう。ウイルスバスターなどのウイルス対策ソフトが TeX Live のインストールを阻害するという問題が報告されているようです。必ず阻害するわけではありませんが、一時的に動作を停止させておいてからインストールすることをオススメします。また、この章では負荷低減のためミラーサイトからのインストール方法を説明します。

#### 2.1.1 Windows の場合

ここでは ISO イメージからのインストールとネットワークインストーラーからのインストールの二種類のインストール方法を説明します。ISO イメージからインストールの方が問題は発生しにくいかもしれません。一方でやってみてダメならもう一方で試してみてください。また、`C:\Users\姓姓 名名` のように、インストールする PC のユーザー名に全角文字や空白などが入るとトラブルの原因となります。ユーザー名を半角のものに変えてからインストールすることをおすすめします。

##### ISO イメージからインストール

1. [ミラーサイト](#) から `texlive.iso` をダウンロード。
2. ダブルクリックすると BD-ROM/DVD-ROM ドライブとしてマウントされる（「セキュリティの警告」が出た場合は「開く」を選択 → エクスプローラーで開く）。
3. 共通事項 4 へ。

## ネットワークインストーラからのインストール

1. ミラーサイトから `install-tl.zip` をダウンロード.
2. `install-tl.zip` を展開.
3. 共通事項 4 へ.

## 共通事項

4. `install-tl-windows.bat` を実行 (青い警告ウィンドウが出たら「詳細情報」→「実行」).
5. TeX Live インストーラが現れたら 「TeXworks をインストール」のチェックを外してからインストール (もし TeXworks が欲しかったらインストールしてもよい). インストールは数時間かかることがあるので注意.
6. インストールできたかどうかチェック.
  - (a) Win+R でファイル名を `cmd` と指定し `cmd.exe` (コマンドプロンプトとも呼ぶ) を開く.
  - (b) `tex -v` と入力し Enter.
  - (c) バージョン情報が出てきたらインストール完了, 出なかったら一度 Path を通してみる.
7. 環境変数 Path の確認.
  - (a) `cmd.exe` を開く.
    - i. `path` と入力し Enter.
    - ii. `C:\texlive\****\bin` (\*\*\*\*には TeX Live のバージョンにあてはまる年が入る)があれば完了. 無ければ 7b へ.
  - (b) Windows の「設定」パネルを開く.
    - i. 「システム」→「バージョン情報」→「システムの詳細設定」→「環境変数」の順に開く.
    - ii. 「システム環境変数」の「Path」をダブルクリック.
    - iii. `C:\texlive\****\bin` があれば完了. 無ければ「新規」で追加し, 7a へ.

### 2.1.2 macOS の場合

macOS の場合は Homebrew でのインストールが簡単です.

```
$ brew install --cask mactex-no-gui  
$ sudo tlmgr update --self --all  
$ sudo tlmgr paper a4
```

## 2.2 使用するエディター

$\text{\TeX}/\text{\LaTeX}$  に対応しているテキストエディターは数多く存在しますが、ここでは Microsoft が開発している Visual Studio Code (VS Code) を紹介します。開発元は Microsoft ですが、Windows だけでなく macOS や Linux でも使用可能です。また、VS Code には豊富な拡張機能が存在しているほか、Git との連携も非常に簡単なため近年非常に人気の高いエディターです。VS Code の詳細な使用方法はここでは割愛しますが、最低限の拡張機能として [\LaTeX Workshop](#) を入れておくとよいでしょう。取り扱う画像ファイルが多くなってきた場合は [Image preview](#) があると便利です。また、VS Code の設定ファイル `settings.json`<sup>1)</sup> でさまざまな設定を書き加えることができます。

## 2.3 pdf ファイルの生成

ここでは実際に pdf ファイル（このテンプレートでは `main.pdf`）を生成する過程を説明します。これまでにある程度  $\text{\LaTeX}$  を使った経験のある方は必要な箇所だけ読めばいい（全部わかつていれば読む必要は無い）と思います。と言っても、 $\text{\LaTeX}$  初心者も全部を読む必要は無く、「一旦このテンプレートで学位論文を書き上げたい」ということを考えている人は第 2.3.1 節の「[Lua\LaTeX](#) の場合（このテンプレートはこちら）」と第 2.3.3 節を読めば大丈夫です。

### 2.3.1 ターミナルでの操作

第 2.3.3 節の `latexmk` を使用すればターミナルでの操作は非常に簡単になりますが、何か問題が発生した際にデバッグを考えるとターミナルでの操作も覚えておく必要があります。実際に pdf ファイルを生成するときは `latexmk` を使用すればいいのですが、まずはどのようなプロセスで実行されているのかを把握しておきましょう。

#### [Lua\LaTeX](#) の場合（このテンプレートはこちら）

この  $\text{\LaTeX}$  テンプレートは [Lua\LaTeX](#) での執筆を前提とし、参考文献は [upBIB\TeX](#) で読み込むようにしています。[Lua\LaTeX](#) は速度がやや遅いものの、高機能で Unicode に対応しているため近年人気が出でてきているモダンな  $\text{\LaTeX}$  です。使い方の詳細は下記のようになります。

1) `settings.json` の一例 <<https://gist.github.com/Yuki-MATSUKAWA/465ecd0ebcbd157e48ac1e3619c9a08c>> を紹介しておきます。 $\text{\LaTeX}$  以外の設定も含まれているので設定の取捨選択は読者の皆さんにお任せします。

**Lua $\text{\LaTeX}$ +upB $\text{\BIBTeX}$** 

```
$ lualatex main  
$ upbibtex main  
$ lualatex main (複数回)
```

まずは主要な  $\text{\LaTeX}$  ソースコードの `main.tex` を  $\text{Lua\LaTeX}$  で読み込むために `lualatex main` とターミナルに入力します。\$は入力しないでください。拡張子の `.tex` は省略可能です。次に、参考文献を読み込むために `upbibtex main` とターミナルに入力します。 $\text{\BIBTeX}$  を使わない処理をしているときはこの操作は不要です。これだけだとまだ  $\text{\LaTeX}$  を使う大きなメリットである相互参照の機能を使えていません。 $\text{\LaTeX}$  で相互参照を有効にするには複数回のコンパイルが必要です。相互参照に失敗した場合やコンパイル回数が足りていない場合は参照箇所が?や??のように表示されるはずです。そのため、 $\text{upB\BIBTeX}$  を読み込んだ後に?や??が消えるまで複数回コンパイルしましょう。これで `main.pdf` を作成できました。

**レガシー  $\text{\LaTeX}$  の場合**

モダン  $\text{\LaTeX}$  とレガシー  $\text{\LaTeX}$  の最大の違いは、pdf ファイルを直接生成できるか否かです。`pLATEX` や `upLATEX` のようなレガシー  $\text{\LaTeX}$  は一度 `dvi` ファイルという中間ファイルを生成し、その後 `dvi` ファイルを `pdf` 等の適切なファイル形式に変換する作業が必要です (`dvipdfmx`)。これから時代はどんどんモダン  $\text{\LaTeX}$  に置き換えられていくと思いますが、まだ対応していない学会・論文テンプレートも多く存在しているのでここで紹介しておきます。また、`pdflATEX` は本来レガシー  $\text{\LaTeX}$  ですが、例外的に直接 `pdf` ファイルを生成でき、国際雑誌論文テンプレートではよく使用されています。ただし、`pdflATEX` は日本語に対応していないため、日本語を使用したい人は  $\text{Lua\LaTeX}$  を使うようにしましょう。どうしても `pdflATEX` で日本語を使用したい（国際雑誌論文執筆の下書き等）場合は第 2.3.2 節を参照してください。

`pLATEX` は日本語に対応した  $\text{\LaTeX}$  として長年愛用されてきましたが、今は  $\text{Lua\LaTeX}$  などに置き換えられてきています。皆さんは使わないようになります。使い方は下記の通り。 $\text{Lua\LaTeX}$  の項目と同様、 $\text{\BIBTeX}$  を使わない場合はそこのコマンドを省略してください。

**pLATEX+pBIBTEX**

```
$ platex main  
$ pbibtex main  
$ platex main (複数回)  
$ dvipdfmx main  
または  
$ ptex2pdf -l main  
$ pbibtex main  
$ ptex2pdf -l main (複数回)
```

上記コマンドの `ptex2pdf -l main` は `platex main` と `dvipdfmx main` を続けて実行するコマンドです。

次に `upLATEX` について説明します。これは `pLATEX` を Unicode に対応させたものとなっており、現在でも広く使われています。そのため、このテンプレートを使用することだけを考える際は不要な情報ですが、念のため載せておきます。`upLATEX` では `upBIBTEX` が使えますが先程と同様、不要な場合は省略してください。

**upLATEX+upBIBTEX**

```
$ uplatex main  
$ upbibtex main  
$ uplatex main (複数回)  
$ dvipdfmx main  
または  
$ ptex2pdf -l -u main  
$ pbibtex main  
$ ptex2pdf -l -u main (複数回)
```

`pLATEX` はレガシー `LATEX` ですが例外的に直接 `pdf` を出力できます (`dvipdfmx` が不要)。日本語には対応していませんが、国際雑誌論文では広く使用されています。どうしても `pdfLATEX` で日本語を使用したい場合は次の第 2.3.2 節を参照。使い方は下記の通り。

**pdfLATEX+upBIBTEX**

```
$ pdflatex main  
$ upbibtex main  
$ pdflatex main (複数回)
```

### 2.3.2 pdfLATEX で日本語を使用する場合

国際雑誌論文等のコンパイルは pdfLATEX が想定されていることがあります。pdfLATEX はレガシー LATEX でありながらも直接 pdf ファイルを生成できることから海外では広く使用されていますが、残念ながら日本語に対応していません。しかし、英語論文の下書きとして日本語を使いたい場合があると思います。その際に、見た目が少し悪くなるものの pdfLATEX で日本語を使用する方法が一応あるのでここで紹介しておきます。

**文書全体で日本語を使用**

```
\usepackage[whole]{bxcjkjatype}
```

まず、LATEX 文書全体で日本語を使用したい場合は上記のコマンドをプリアンブルに書きます。これで文書全体で日本語の使用が可能になります。ただし、前述の通り見た目が悪くなるので下書き用（後で英語に変更する用）として使用してください。

**文書の一部分で日本語を使用**

プリアンブルに記載

```
\usepackage{CJKutf8}
```

本文中に記載

```
\begin{CJK}{UTF8}{ipxm}
```

日本語

```
\end{CJK}
```

次に、文書全体ではなく一部分でのみ日本語を使用したい場合のコマンドは上記のようになっています。まず、\usepackage{CJKutf8} というパッケージを読み込むことで日本語を使用できるようにします。厳密には日本語だけでなく、中国語 (Chinese)、日本語 (Japanese)、韓国語 (Korean) の組版規則に対応させるためのパッケージとなります。次に本文中の日本語を使いたい箇所を \begin{CJK}{UTF8}{ipxm} と \end{CJK} で囲ってあげれ

ばそこでは日本語を使えるようになります。米国物理学協会（American Institute of Physics, AIP）が発行している雑誌論文（Physics of Fluidsなど）は著者の氏名で英語表記以外に漢字等の表記を併記することが可能になっています。このようなときにこのコマンドを使ってあげるとよいでしょう。また、日本語を使う箇所がもう少し長い場合はプリアンブルで `\newcommand*{\Ja}[1]{\begin{CJK}{UTF8}{ipxm}#1\end{CJK}}` のようにコマンドを作つてあげてもいいかもしれません。

### 2.3.3 latexmk を使う方法

L<sup>A</sup>T<sub>E</sub>X 関連のファイルが変更されるたびに第 2.3.1 節で紹介した操作を毎回行うのは非常に面倒です。そこで `latexmk` という機能を使って簡略化しましょう。`latexmk` を使うと、このリポジトリ内に入っている `latexmkrc` というファイル<sup>2)</sup>を呼び出し、実行したいコマンドを一回の操作で実行してくれます。`latexmkrc` は `main.tex`（主要な L<sup>A</sup>T<sub>E</sub>X コード）と同じ階層に用意しておいてください。あとはターミナル上で `latexmk main` と打てばすべて実行してくれます。

#### latexmk を使用

```
$ latexmk main
```

これで随分楽になったと思いますが、VS Code を使っている皆さんにはもっと楽にできます。私が使っている `settings.json` の中で L<sup>A</sup>T<sub>E</sub>X のビルト時に `latexmk` で実行するように設定してあるので、Windows の場合は Ctrl+Alt+B で同様の操作を行ってくれます。`settings.json` の設定を変更して自動コンパイルにすることもできます。また、Windows で pdf ファイルのプレビューを見たい場合は Ctrl+Alt+V の操作で表示できます。Ctrl キーを押しながらマウスでプレビューをクリックすると該当箇所のソースコードに飛べるのも便利な機能です。

### 2.3.4 クラウド上で使う方法

第 2.1 節で環境構築の方法を述べました。本音としては L<sup>A</sup>T<sub>E</sub>X の全ユーザーが自身の PC にローカルの L<sup>A</sup>T<sub>E</sub>X 環境を整えてほしいのですが、環境構築に手間がかかる不便さもあるため、ここでは TeX Live 等のインストールをせずにクラウド上で L<sup>A</sup>T<sub>E</sub>X 環境構築方法を説明します。クラウド上で L<sup>A</sup>T<sub>E</sub>X を使用できるツールとしては Cloud LaTeX や Overleaf といったものが有名で、私は Overleaf をよく使っているのでここでは Overleaf の説明をします。

Overleaf は複数のユーザーによる（同時）共同執筆も可能となっており、Overleaf 自体が

2) 拡張子はつけないでください。

Git と同様の役割を担っているため大変便利です。もちろん Git/GitHub との連携も可能となっています。

後で書きます。

# 第3章

## LATEX\ の基本

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

### 3.1 LATEX\ とは何か

$$Re_{\text{in}} = \frac{u_{\text{in}} h}{\nu}, \quad Re_{\text{out}} = \frac{u_{\text{out}} h}{\nu}, \quad Re = \frac{u_0 h}{\nu} \quad (3.1)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \quad (3.2)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \quad (3.3)$$

# 第4章

## 図表の配置

LATEX で図や表を挿入するときのコマンドは初心者には覚えにくいです。また、インターネットで検索したものを継ぎ接ぎした結果、何が何だかよくわからないコードができあがるということがよく起きるのでこのファイルからコピーアンドペーストすれば問題ないようにしておきました。この章ではわかりやすくするために図表のキャプションを全て日本語で書いていますが、実際に学位論文を書く際は全て英語で書いてください。

### 4.1 図の配置

#### 4.1.1 図を1枚だけ配置する方法

ここでは図を1枚だけ配置する方法を紹介します。図を配置するときは `figure` 環境で図を自動配置し、`\includegraphics` で図を挿入します（図 4.1 のコードを参照）。`figure` のオプション [] の中にある文字は出力する場所を示します。

- t ページ上部 (**top**) に図を出力
- b ページ下部 (**bottom**) に図を出力
- p 単独ページ (**page**) に図を出力
- h できるだけその位置 (**here**) に図を出力
- H 必ずその位置 (**Here**) に図を出力 (`float` パッケージを必要とする)

学位論文中の図は原則ページ上部に配置するのでこの `.tex` ファイル中では `[tp]` に設定しています。皆さんはこのままコピーしてください。`\columnwidth` は現在のコラムのテキスト幅を指しており、`[width=0.5\columnwidth]` と設定することで、テキスト幅の半分の横幅で図を挿入できます。図の大きさの指定に関してよく使うコマンドをテキストボックスにまとめています。`\textwidth`, `\columnwidth`, `\ linewidth` はよく似たコマンドですが、二段組の論文の場合はそれぞれの段の列幅が `\columnwidth` になり、`\ linewidth` はリストなどの環境下での行の長さで臨機応変に対応します。これらは文書内のある長さに対して相対的に図の大きさを決定する方法でしたが、`width=25mm` のように絶対的な長さも指定できます。

\centering は図を中央寄せするコマンドです<sup>1)</sup>.

また、コードにもあるように \label{} コマンドを挿入することでラベルを設定できます。ここでは \label{fig:one\_figure} としており、ラベル参照時に図であることがわかるよう fig: を入れています。ご自身の論文の内容に合わせてキャプションやラベルは変更してください。文章中で引用する際は 図~\ref{fig:one\_figure} のように書きます。すると図 4.1 のように出力されます。ハイパーリンクも埋め込まれているので該当する図が遠く離れた位置にあっても便利です。ここで「図」と番号の間にチルダ ~ を入れているのはここでの改行を防止を目的としています。

### \includegraphics で図の大きさの指定によく使うコマンド

#### 【指定するもの】

コマンド	意味	使用例
width	画像の幅	width=0.5\textwidth
height	画像の高さ	height=0.1\textheight
scale	画像のスケール	scale=0.5

#### 【長さに関するコマンド】

コマンド	意味
\textwidth	テキストエリアの幅
\textheight	テキストエリアの高さ
\columnwidth	テキスト列の幅
\ linewidth	現在の環境内での行の長さ

## 4.1.2 図を複数枚配置する方法

関連する図（ここではそれぞれの図を「サブ図」と呼称します）を複数枚配置するときは subcaption を使いましょう<sup>2)</sup>。このテンプレートでは settings.sty 内で読み込んでいます。文章中では subfigure 環境に入れて並べます。例えば 2 枚の図を横に並べて配置したいときは図 4.2 のようになります。ここでは \hfill を使って図と図の間の空白を設定していますが、\hspace{3mm} のように設定しても構いません。 \hspace{3mm} の場合、水平方向に 3 mm の空白ができます。3 枚のサブ図を横に並べたいときも同様で、図 4.3 のようになります。関連するサブ図を横だけでなく縦方向にも配置したいときは、図 4.4 のように横並びの

1) 図などを中央寄せする際に \begin{center} で始まる center 環境を使うのは非推奨です。

2) subfigure や subfig は古いので非推奨です。



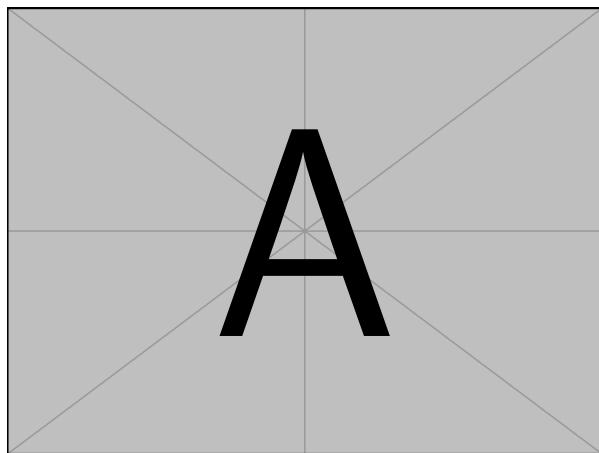
Fig. 4.1 1枚の図.

\columnwidth の合計が大きくなりすぎると自動的に縦に配列してくれます。ここでは縦方向のスペースを確保するために \vspace{5mm} を挿入しています。

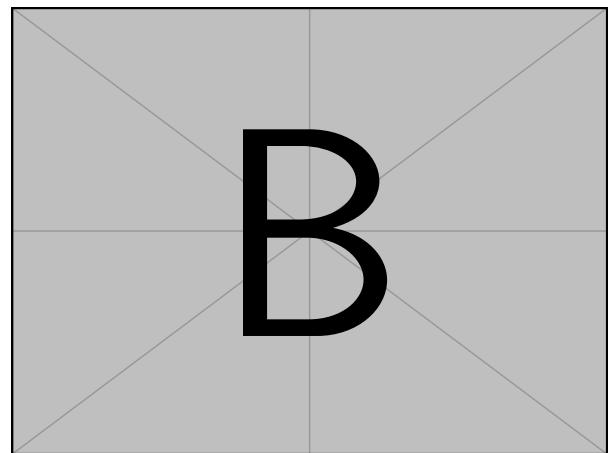
### 図のラベルの参照方法

入力	出力
\ref{fig:one_figure}	4.1
\ref{fig:two_figures}	4.2
\ref{subfig:two_figures_a}	4.2a
\ref{fig:two_figures}(\subref{subfig:two_figures_a})	4.2(a)
(\subref{subfig:two_figures_a}, \subref{subfig:two_figures_b})	(a, b)
(\subref{subfig:four_figures_a}--\subref{subfig:four_figures_c})	(a-c)

また、subfigure 環境を使うことでそれぞれのサブ図にラベルを付けることができます。参照時には \ref{fig:two\_figures} と入力すると 4.2 のように図全体の番号のみ、\subref{subfig:four\_figures\_a} と入力すると a のようにサブ図の番号のみ出力されます。図 4.2 のように出力したい場合は図 ~\ref{fig:two\_figures} とすればよいですが、仮に図 4.2(a) のように出力したい場合は図 ~\ref{fig:two\_figures}(\subref{subfig:two\_figures\_a}) とします。このとき、

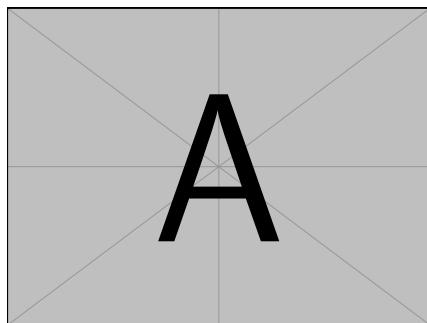


(a) 左の図.

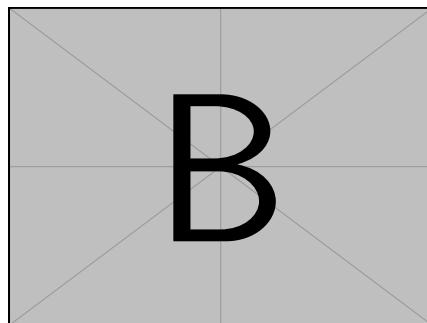


(b) 右の図.

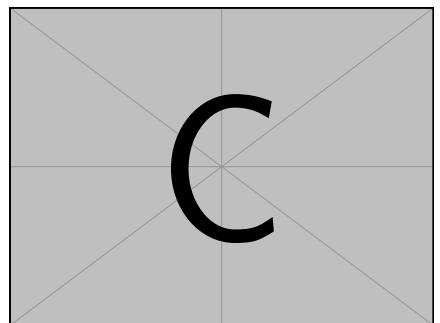
Fig. 4.2 左右の図.



(a) 左の図.



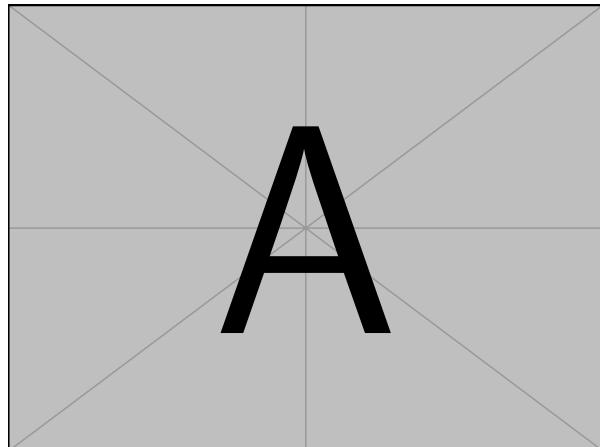
(b) 中央の図.



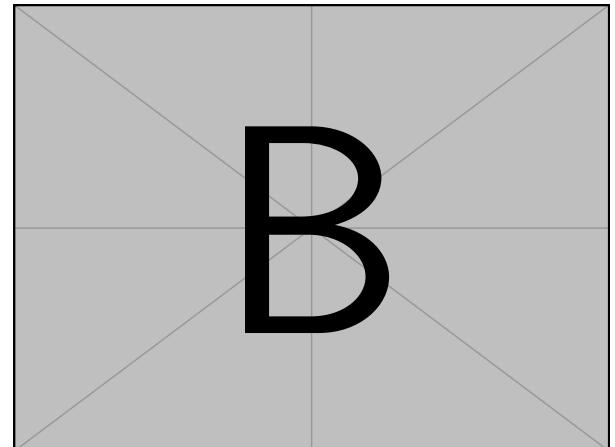
(c) 右の図.

Fig. 4.3 3枚の図.

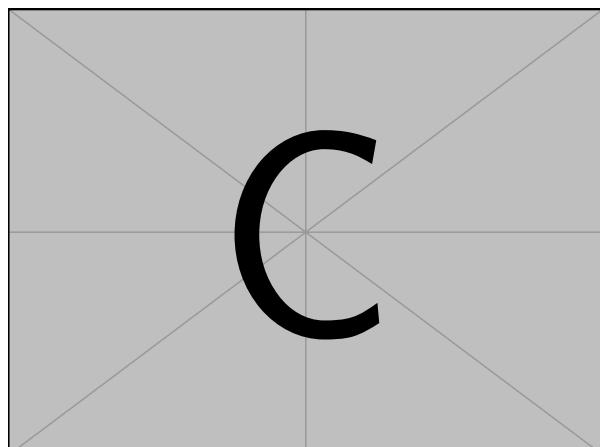
\subref{} 前後の括弧 () を忘れないでください。仮に \ref{subfig:two\_figures\_a} のようにサブ図を \ref{} コマンドで直接指定してあげると 4.2a のように図番号とサブ図番号が括弧無しで出力されます。括弧をデフォルトで出力するような設定もできますが、図 4.2(a, b) や図 4.4(a–c) のように複数のサブ図を指定するときに不便なので括弧を外してあります。もしデフォルトで括弧を出力する設定に変更したい場合は settings.sty 内でコメントアウトしている \renewcommand{\thesubfigure}{(\alph{subfigure})} を有効化してください。



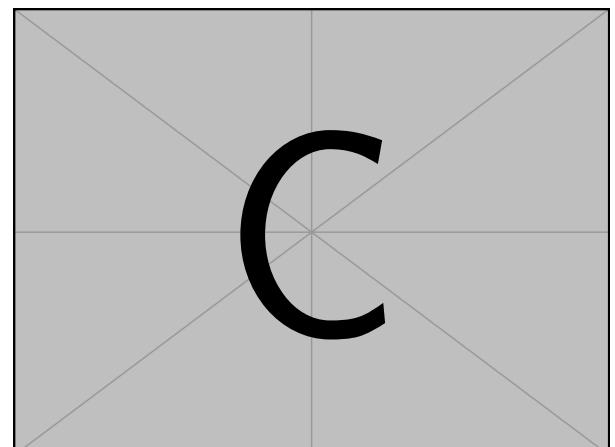
(a) 左上の図.



(b) 右上の図.



(c) 左下の図.



(d) 右下の図.

Fig. 4.4 上下左右に4つ配置した図.

### 4.1.3 画像のファイル形式

画像形式は大きく分類するとラスター画像とベクター画像に分類できます。結論から先に述べると、**ラスター画像であれば JPEG か PNG、ベクター画像であれば PDF を使用してください。**

- ラスター画像：小さな正方形（ピクセル、画素）を大量に組み合わせて作り上げた画像。ラスター画像を拡大するとピクセルの存在を確認できる。ラスター画像の例は以下の通り。
  - GIF (Graphics Interchange Format)：拡張子は .gif で、256 色以下の画像を扱える可逆圧縮形式ファイル。使用できる色は少ないものの、アニメーションにも対応していることから現在でも使う機会が多い。
  - JPEG (Joint Photographic Experts Group)：拡張子は .jpeg や .jpg で、最大 24 ビット（約 1677 万色）の色に対応している非可逆圧縮形式ファイル。
  - PNG (Portable Network Graphics)：拡張子は .png で、JPEG と同様 24 ビットの色に対応している可逆圧縮形式ファイル。透過処理にも対応している。
- ベクター画像：円や直線などを数式的に処理することで作り上げた画像。どれだけ拡大しても明瞭なまま。ベクター画像の例は以下の通り。
  - PS (PostScript)：拡張子は .ps で、Adobe が 1984 年に開発したページ記述言語で組まれた画像形式。
  - EPS (Encapsulated PostScript)：拡張子は .eps で、PostScript の後継となる画像形式（カプセル化された PostScript）。バウンディングボックスを読み込むことで描画領域を確保する。
  - PDF (Portable Document Format)：拡張子は .pdf で、環境に左右されず、ほぼ同様の見た目で画像や文書を閲覧できる。一般的な用途では最も主流なベクター形式。

ラスター画像かベクター画像かという観点では、論文中の画像はできるだけベクター画像の方がいいです。これは上記説明にも書いたように、ベクター画像は内部で数式処理をしているためいくら拡大しても解像度が落ちず明瞭なままだからです。ただし、これは一般的なグラフや簡単なカラーマップ限定の話です。複雑なカラーマップをベクター画像にするとファイルサイズが膨大になり、画像を開くだけでも一苦労です。このような場合には諦めてラスター画像にしましょう。

また、一昔前の  $\text{\LaTeX}$  では画像の挿入と言えば EPS ファイルでした。しかし、現在の  $\text{\LaTeX}$  事情では EPS ファイルの使用は非推奨です。本来  $\text{\TeX}$  エンジンは EPS ファイルを直接処理

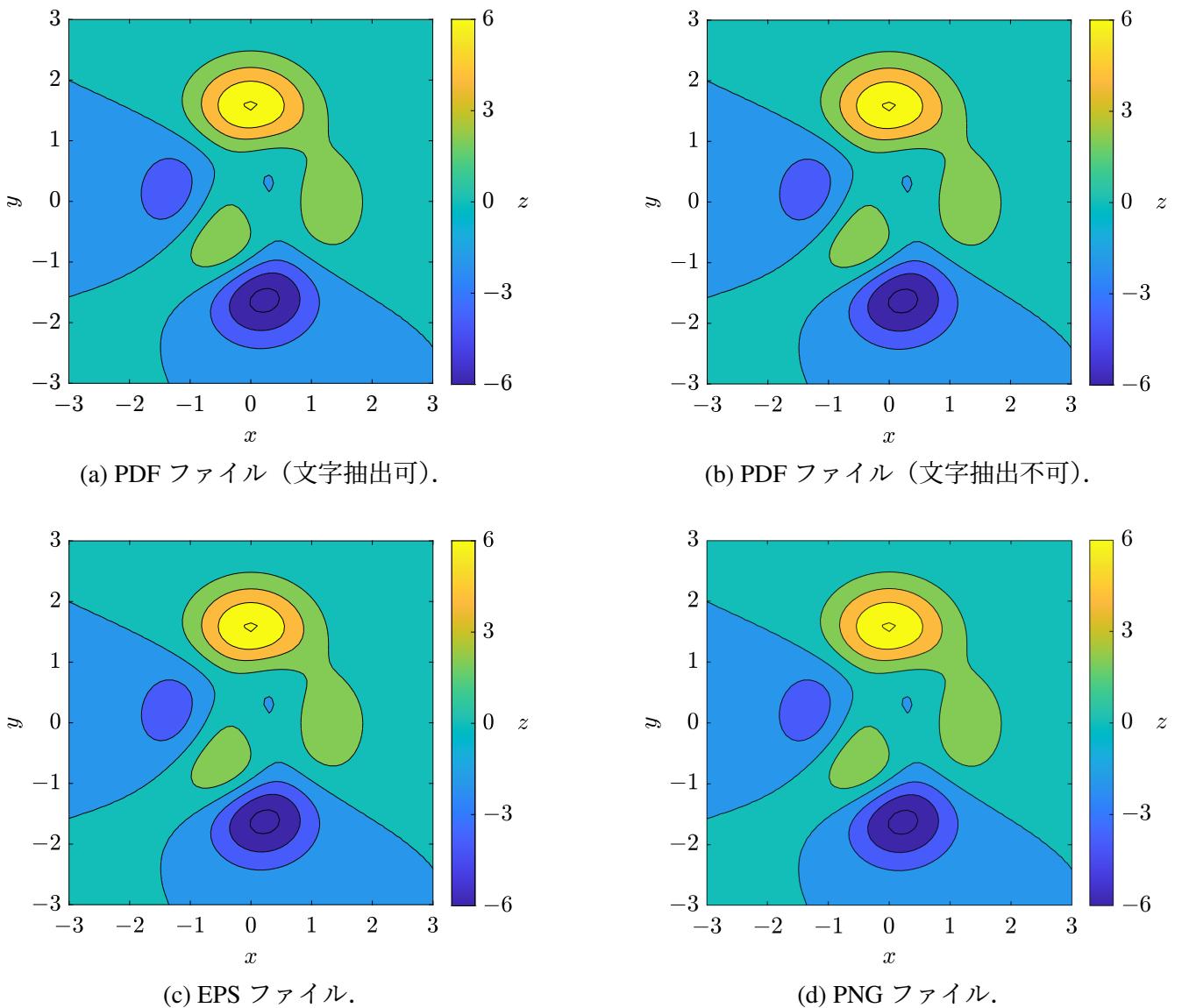


Fig. 4.5 画像形式ごとの比較.

することができず、Ghostscript という PostScript 言語のインタープリターを経由しなければいけません。したがって、最初から PDF で挿入する方がよいというわけです。また、バウンディングボックスの調節がうまくいかず、EPS で挿入すると画像がずれるという問題が生じることがあります。現代に生きる皆さんは EPS ではなく PDF を使いましょう。

それでは実際にいくつかの画像形式を比較してみましょう。図 4.5 では (a, b) PDF ファイル, (c) EPS ファイル, (d) PNG ファイルを並べて比較しています。パネル (a–c) はどれもベクター画像なのでいくら拡大しても明瞭なままであります。一方のパネル (d) を拡大するとラスター画像なので小さな正方形で構成されていることが確認できます。これがベクター画像とラスター画像の違いです。それではパネル (a) と (b) の違いは何でしょうか。どちらも PDF 形式ですよね。main.pdf を開きながら Ctrl+A をしてください。パネル (a) は文字を抽出で

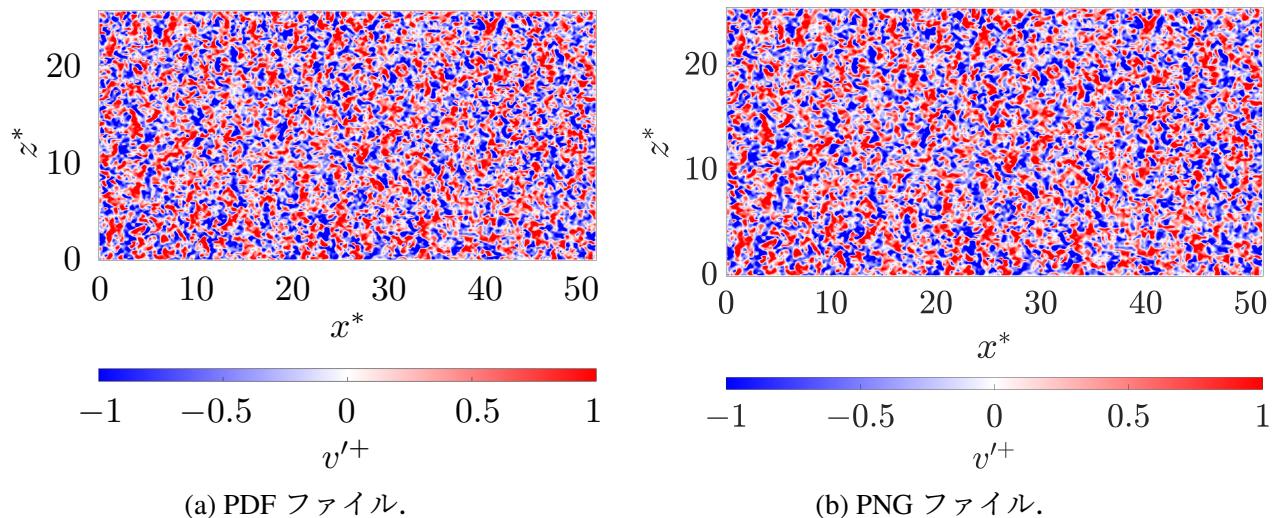


Fig. 4.6 ファイルサイズが大きい画像の比較.

きますが、(b) は文字を抽出できません。皆さんが論文を書く際は (a) 文字抽出が可能な PDF を使用するのが理想です。

ただし、全ての画像をベクター画像にするのは難しい場合もあります。図 4.5 の場合はファイルサイズの小さいカラーマップだったのでベクター画像として貼っても問題ありませんが、もっと複雑な模様だととても扱えるようなファイルサイズではありません。図 4.6(a) はカラーマップだけラスターとし、文字情報は全てベクターで抽出できるようになっています（ファイル自体は PDF）。パネル (b) は全てラスターの画像（PNG）なので文字を抽出することはできません。ただ、図 4.6(a) のような図を作るのは少し技術を要するので、重い画像ファイルの場合は (b) のようなラスター画像でもいいでしょう。

## 4.2 表の配置

次に表の作り方を説明します。正直、 $\text{\LaTeX}$  環境での表作成は少々面倒です。特に表のセルの数が多くなると行をいくつも増やさなければいけないのでかなりの労力がかかります。簡単に  $\text{\LaTeX}$  の表を作ってくれるツールとして、Tables Generator<sup>3)</sup> があります。また、2023年9月27日に Overleaf に入ったアップデートで直感的な表の作成が可能になりました。Overleaf の表作成機能はかなり便利だと思うので、ローカルで  $\text{\LaTeX}$  文書を書いているときに表作成時だけでも Overleaf を立ち上げるとストレス無く表を作れると思います。

ここでは `tabular` 環境を用いた表作成の方法と `tblr` 環境を用いた表作成の方法の両方を記載します。`tabular` 環境は古くから存在している表作成の方法ですが、カスタマイズ性が

3) Tables Generator, <<https://www.tablesgenerator.com/>>

Table 4.1 tabular 環境を用いた表のサンプル。

学会名	会員種別	年会費
実在する学会		
日本機械学会	学生員	4,800 円
日本流体力学会	学生会員	5,000 円
日本伝熱学会	学生会員	4,000 円
実在しない学会		
日本架空学会	小学生会員	-8,000 円
	中高生会員	-5,000 円
	大学生会員	-2,000 円
	名誉学生会員	$6.02 \times 10^{23}$ 円

低く、非常に使いにくいです。しかし、tabulararray パッケージ<sup>4)</sup>でサポートされているtblr 環境は非常にカスタマイズ性が高く、tabular 環境で非常に難しかったセル結合も容易に行えます。tbluar 環境は現在でも広く使われているためこのテンプレートでも説明しますが、皆さんが論文を書くときは是非tblr 環境を使ってみてください。

### 4.2.1 tabular 環境を用いた表

日本機械学会が推奨する表形式を満たしたサンプルを表 4.1 に示します。コードは表 4.1 の該当箇所を確認してください。表を作成するときは table 環境の中に tabular 環境を作ります。table 環境は、表やキャプション、ラベルを全て含めた表全体の制御を行い、tabular 環境は表の各セルの制御を行います。table 環境は figure 環境と同様、h, t, b, p, H による位置制御を行います。表の場合は図と異なり、キャプションは上に付けます。tabular 環境内でセルの文字揃え位置制御は lcr で行います。

- l 左揃え (left)
- c 中央揃え (center)
- r 右揃え (right)

表 4.1 の場合は {l|c|r} としています。この場合、左の列から左寄せ・中央寄せ・右寄せになります。縦棒（パーティカルライン）|は表の縦罫線を入れる場所を示しています。この場合、1列目と2列目の間、2列目と3列目の間に縦罫線を入れます。行の区切りは \\、列の

4) L<sup>A</sup>T<sub>E</sub>X3 を利用した新しいパッケージです。

区切りは & です。横罫線を引くときは `\hline` を使います。日本機械学会のテンプレートの表では一番上の横罫線は 2 本なので `\hline\hline` としています。

次に表のセル結合について説明します。行方向のセル結合を行う際は `\multicolumn{結合する列数}{揃え位置}{セルの中身}` を使います（結合するセルは列）。表 4.1 の 2 行目で使用している `\multicolumn{3}{c}{実在する学会}` は「横並びの 3 つのセルを結合し、『実在する学会』という文字列を中心揃えで配置する」命令です。1 行 1 列目では `\multicolumn{1}{c}{学会名}` としていますが、これは本来左揃えになっている 1 列目を、このセルだけ例外的に中央揃えにするために使用しています。面白い使い方ですね。列方向のセル結合を行う際は `\multirow{結合する行数}{幅}{セルの中身}` を使います（結合するセルは行）。表 4.1 の 7 行 1 列目で使用している `\multirow{4}{*}{日本架空学会}` は「縦並びの 4 つのセルを結合し、『日本架空学会』という文字列を幅指定なしで配置する」命令です。列方向のセル結合を行う際は横罫線を消す必要があります。`\cline{x-y}` という命令を使うと、 $x$  列目から  $y$  列目に横罫線を入れるコマンドです。`\hline` がその行全体に横罫線を入れるのに対して `\cline{}` はその行に部分的な横罫線を入れるコマンドです。横罫線を消すことでセル結合したような出力を得られます。

#### tabular 環境でセル結合する際に使用するコマンド

<code>\multicolumn{結合する列数}{揃え位置}{セルの中身}</code>	列のセル結合
<code>\multirow{結合する行数}{幅}{セルの中身}</code>	行のセル結合
<code>\cline{start-end}</code>	一部の横罫線のみを表示

また、表の中では数式を使用することも可能です。やりがちなミスとして、負の数をセルに入れるとときに数式環境 `$$` に入れ忘れて、マイナスがハイフンとなって出力されているものがあります<sup>5)</sup>。

#### 4.2.2 tabular 環境を用いた表

表 4.2 は `tblr` 環境を用いた表のサンプルです。見た目が表 4.1 と同じになるようにしています。基本的な使い方は `tabular` 環境に準じています。大きな違いは罫線の扱いです。`\begin{tblr}` の直後で `hlines` を入れると全ての横罫線が、`vlines` を入れると全ての縦罫線が表示されます。また、セル結合の方法も異なります。表 4.2 の 2 行目の `\SetCell[c=3]{c} 実在する学会` は「横並びの 3 つのセルを結合し、『実在する学会』という文字列を中心揃えで配置する」命令です。1 行 1 列目では `\SetCell[c]{c} 学会名` とするこ

5) 例えば `-100` と表示するには `$-100$` と入力する必要があります。 `-100` だと `-100` と表示されます。

Table 4.2 `tblr` 環境を用いた表のサンプル。

学会名	会員種別	年会費
実在する学会		
日本機械学会	学生員	4,800 円
日本流体力学会	学生会員	5,000 円
日本伝熱学会	学生会員	4,000 円
実在しない学会		
日本架空学会	小学生会員	-8,000 円
	中高生会員	-5,000 円
	大学生会員	-2,000 円
	名誉学生会員	$6.02 \times 10^{23}$ 円

とでこのセルだけ中央揃えに変えています。7行1列目の `\SetCell[r=4]{1}{日本架空学会}` 日本架空学会は「縦並びの4つのセルを結合し、『日本架空学会』という文字列を左揃えで配置する」命令です。

#### tblr 環境でセル結合する際に使用するコマンド

<code>\SetCell{c}</code>	そのセルだけ中央揃えに変更
<code>\SetCell[r=3]{c}</code>	3行をセル結合し中央揃え
<code>\SetCell[c=3]{c}</code>	3列をセル結合し中央揃え
<code>\SetCell[r=2,c=3]{c}</code>	2行3列をセル結合し中央揃え

# 第 5 章

## BIBTEX による参考文献一覧の出力

この章では参考文献の一般的な記載方法, LATEX での処理方法について説明します。第 5.1 節はこのテンプレートに限らない、一般的に学術論文等で参考文献を載せる際に注意すべき点をまとめているので読み飛ばしていただいて構いません。

### 5.1 参考文献の記載時の一般的な注意事項

論文執筆の際には多くの文献を引用することになります。読者に正しい情報を提供するのももちろんのこと、先人たちの業績を認め、評価するという観点でも文献を引用する際は細心の注意を払いましょう。

#### 5.1.1 引用方式

参考文献の引用方法は Harvard 方式と Vancouver 方式に大別できます。

- Harvard 方式

- 本文中での引用はいわゆる author-year 方式。「著者名」と「発行年」を記載する。
- 本文中での引用は苗字だけでの記載が多い。引用例：
  - \* 著者 1 名 : [Reynolds \(1883\)](#)
  - \* 著者 2 名 : [Schmid and Henningson \(2001\)](#)
  - \* 著者 3 名以上 : [Berghout et al. \(2020\)](#)
- et al. はラテン語で「その他」を意味する et alii の略。Italic 体で書くことが多い。
- 論文末尾の文献リストは著者名のアルファベット順でソート。

- Vancouver 方式

- 本文中での引用は番号。
- 本文中での引用例 : ~が明らかになっている<sup>[1,2]</sup>.
- 論文末尾の文献リストは本文での登場順でソート。

日本機械学会の論文執筆規定では Harvard 方式になっているため、この学位論文テンプレートも Harvard 方式を採用しています。

### 5.1.2 文献リストの作り方

後で書きます。sentence caseとか。

### 5.1.3 文献の正しい引用の仕方

後で書きます。孫引きとか。版・刷とか。

## 5.2 BibTeXの使い方

L<sup>A</sup>T<sub>E</sub>Xでは文献リストを作る方法として `thebibliography` 環境の中で `\bibitem` コマンドを使用する方法があります。しかし、この方法は文献リストを人間が全て手打ちで入力しなければいけません。引用する文献が片手で数え切れるくらいの数であれば全て手打ちで文献リストを作ってもいいかもしれません、学術論文や学位論文になると人力で文献リストを作るのは時間の無駄ですしうまくやれません<sup>1)</sup>。

そこで、BibTeXを使用した参考文献リストの作成方法を説明します。BibTeXを使えばユーザーが作成した `bib` ファイルを読み込んで自動で文献リストを作ってくれます<sup>2)</sup>。

### 5.2.1 jsme.bstについて

第 5.1 節で文献の一般的な引用の仕方を説明しましたが、学会やジャーナルによって細かいルールは異なります。それぞれの論文での引用ルールに則った出力を得るために必要なものが BibTeXスタイルファイル、`bst` ファイルです。BibTeXを走らせるときは `bst` ファイルを読み込んで文献リストの出力方法を決めます。有名な `bst` ファイルとして `jplain bst` や `jecon bst` があります。

東京理科大学創域理工学部機械航空宇宙工学科の卒業論文では、参考文献一覧および本文中の引用に関して一般社団法人日本機械学会の論文執筆テンプレートの書き方に沿って記載するよう決められています。しかし、日本機械学会から公式な BibTeXスタイルテンプレートは配布されていません。そこで、塙原研究室所属の学生が日本機械学会の参考文

1) 学会の講演論文テンプレート等では、人力で文献リストを作る方法が採用されているものが結構あります。`\bibitem` コマンド等を使って自力でリストを作成する方法は大抵の L<sup>A</sup>T<sub>E</sub>X入門書に記載があるのでそちらを参考にしてください。

2) BibTeXは現在でも広く使われていますが、最近は BibTeXの後継として `biblatex` が徐々に普及してきています。この学位論文テンプレートでは後述の `jsme.bst` を使用しているため BibTeXの内容に限定して記載しています。将来的には `biblatex` に置き換えたと考えています。

献の書き方を再現した「非公式の」BIBTEXスタイルテンプレート<sup>3)</sup>を開発し、GitHubで公開しているのでこれを使用します。使用方法は一般的なBIBTEXと同様ですが、詳細な説明書([JSME-template1.pdf](#))がリポジトリ内にあるので何か問題があった場合はそれを読むようにしましょう。日本機械学会の規定通りの文献出力を得るにはjsme.bstを使用すれば大丈夫ですが、TUS-ME\_thesis\_templateリポジトリ内には最初からjsme.bstが入っているのでこれを読んでいる皆さんが新たにjsme.bstファイルをJSME-bstリポジトリから移してくる必要はありません。

### 5.2.2 bibファイルについて

BIBTEXは自動で文献リストを作ってくれるとは言ったものの、書誌情報は与えてあげないといけません。bibファイルには自分が引用する書誌情報を記載します。bibファイルの書き方はjsme.bst内の[JSME-template1.pdf](#)で詳細に書いてあるのでそちらをよく読んでください。BIBTEX初心者にとっても痒い所に手が届くように書かれています。ただ、基本的な内容だけここにも書いておきます。

bibファイルに入力する書誌情報は次のような構造になっています。

#### bibファイル内の書誌情報の構造

```
@エントリー名{参照キー,  
    フィールド1 = {},  
    フィールド2 = {},  
    フィールド3 = {}  
}
```

だいたいの雑誌論文のウェブサイトではBIBTEX形式で書誌情報を出力できる機能があるのでそこからbibファイルをダウンロードします。もちろん、ダウンロードしたbibファイル自分で書き換えることもできますし、自分で一からbibファイルを作成することも可能です。文献を本文中で引用する際は\citet{Matsukawa:PoF2022}のように書きます。このときのMatsukawa:PoF2022が参照キーです。参照キーの書き方に特に規則は無く、半角カンマ以外の半角記号も使用可能です。ただ、自己の中でマイルールを設けておくと引用する際に楽です。

3) JSME-bst, <<https://github.com/Yuki-MATSUKAWA/JSME-bst>>

# 第6章

## 先生や先輩に添削してもらうときの注意点

### 6.1 latexdiff

### 6.2 latexdiff-vc

# 第7章

## さらに詳しい情報が欲しい人は

### 7.1 論文の書き方に関する情報

- 理科系の作文技術
- 日本語の作文技術

### 7.2 $\text{\TeX}/\text{\LaTeX}$ に関する情報

texdoc

#### 7.2.1 書籍

- 美文書作成入門
- 吉永本

#### 7.2.2 インターネット上の情報

# 謝辞

謝辞は非常に重要な章です。学位論文執筆にあたりお世話になった人は必ず書きましょう。私は基本的に謝辞は自由に書いてほしいと思っていますが、明らかな間違いを書いてくる人を結構見てきました。これまで添削してきた論文の中でよく見た間違いや書き方の例などをここでまとめておきます。

まずは全体に共通する注意事項です。

- 論文本体は基本的に常体（だ・である）で書きますが、謝辞に限り敬体（です・ます）で書いても構いません。
- 氏名の書き間違い等には十分気をつけてください。間違えると大変失礼です。心配な場合は大学や研究室の web ページ等からコピーアンドペーストするとよいでしょう。
- また、所属名の間違いにも気をつけましょう。卒業論文執筆時のときと修士論文執筆時で所属名が変わっていることもあります（大学の学部再編等の事情による）。
- 基本的な日本の大学教員の職階（役職名）は上から順に以下の通りです。
  1. 教授
  2. 准教授
  3. 講師
  4. 助教
  5. 助手

大学によっては上記職階の一部が設置されていない場合があります。また、よくある間違いとして、准教授や助教を「助教授」と書くことが挙げられます。助教授はかつて実際に存在した職階ですが、2007 年の学校教育法の改正に伴い、准教授へと名称が変更されました。また、助教が助教授の略称だと勘違いしているケースもよく見ます。

- 普段大学で○○教授と呼ぶことはありませんが、学位論文の謝辞では「フルネーム + 役職名」で記載しましょう。○○教授や○○准教授と比較して○○講師や○○助教という言い方はあまり聞きませんが、統一して役職名で書いてください。その際、役職名を間違えると大変失礼です。卒業論文執筆時に准教授だった先生が修士論文執筆時には教授に昇進していることもあります。卒論の謝辞を使いまわすのではなく、よく確認しておくように。

- 修士課程や博士後期課程における「課程」を「過程」と書く間違いをよく見るので気をつけましょう。
- また、大学の場合は「卒業」で大学院の場合は「修了」です。「修了」を「終了」と書いてしまうミスもたまに見ます。
- 謝辞はどうしても同じような文言や文末表現が連続しがちです。しかし、感謝を伝えるためにも表現を少しづつ変えてください。

次に、おおまかな書く順番とそれぞれの注意事項は下記の通りです。必ずこの順番を守らなければいけないというわけではありませんが、このような順であることが多いです。

#### 1. 指導教員（主査）

- 書き方の例：指導教員である東京理科大学創域理工学部機械航空宇宙工学科の○○教授は……
- 大学によっては教員の所属が大学院の場合もありますが、東京理科大学の場合は学部です。
- よくある間違いとして「指導教官の……」と書くものです。教官という言い回しは昔の国立大学の教員のものです。正しくは「指導教員」です。
- 稀に実質的な指導教員と主査が異なる場合があります。

#### 2. 共同研究者

- 書き方の例：例：○○研究所の○○博士は……
- 共同研究者は大学教員の場合もあれば大学以外の研究機関の職員の場合もあります。所属名と役職名をよく確認しておきましょう。

#### 3. 副査の先生方

- 書き方の例：○○教授と○○准教授には本論文の副査を引き受けいただき……
- 卒業論文の場合は副査がありませんが、修士論文の場合は2名の副査があります。

#### 4. その他お世話になった学生以外の先生方・研究者

- 書き方の例：本研究室所属の○○博士研究員は……
- 研究室に博士研究員（いわゆるポスドク）や直接の自分の指導教員ではない助教や秘書の方がいる場合も忘れずに書いておきましょう。

#### 5. 研究助成元や計算機センター等

- 書き方の例：直接数値計算の一部は東北大学サイバーサイエンスセンター大規模計算システム AOBA を利用しました。
- 研究遂行にあたり助成等を受けた場合に記載します。奨学金をもらっている場合もここに書きましょう。

## 6. 研究室メンバー

- 書き方の例：博士後期課程1年の○○氏は.....
- 研究室メンバー全員を列挙しなければいけないわけではありません。論文執筆にあたって本当に貢献した人を書いてください。全員の貢献があっての学位論文だと思うのであれば全員書いてもいいと思います。
- 並べ方は基本的に貢献度順です。自分と同じ研究班の人をまずは優先的に書きましょう。
- 貢献度が同じだと判断した場合は上からの学年順に並べましょう。
- 貢献度と学年が同じ人は五十音順に並べましょう。
- 日本学術振興会特別研究員に採用されている博士後期課程学生がいる場合は肩書きとして併記することが多いです。

## 7. 家族

- 書き方の例：最後に、私の大学院進学に対して理解を示し、常に私を気にかけていたいた祖母と両親に感謝申し上げます。
- 自分を応援してくれた家族への感謝も忘れずに。
- 残念ながら自分の卒業・修了を見届けられずにご家族が亡くなってしまう場合もあります。その際は「天国で見守ってくれている○○」などと書くことが多いです。
- なお、家庭環境等の事情により家族を記載したくない場合はこの限りではありません。心の中で感謝しておきましょう。

# 文献

- Berghout, P., Dingemans, R. J., Zhu, X., Verzicco, R., Stevens, R. J. A. M., van Saarloos, W. and Lohse, D., Direct numerical simulations of spiral Taylor–Couette turbulence, [Journal of Fluid Mechanics](#), Vol. 887 (2020), A18.
- Reynolds, O., An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels, [Philosophical Transactions of the Royal Society of London](#), Vol. 174 (1883), pp. 935–982.
- Schmid, P. J. and Henningson, D. S., Stability and transition in shear flows, [Springer New York](#) (2001).

# 付録 A

## 修士課程における研究成果

### 国際学術雑誌論文（査読あり）

1. Ridai, T. and Kikai, H., History of computational fluid dynamics, [Journal of Rikadai Dynamics](#), Vol. xx, No. x (20xx), xxxxxx.

### 報告書

1. 理大太郎, 機械花子, 数値流体力学の歴史, [日本数値流体力学研究所広報誌](#), Vol. xx, No. x (20xx), pp. xx–xx.

### 受賞

1. **Best Paper Award**, 22nd Tokyo University of Science Conference (TUSC22), 23–26th Sep. (20xx).

### 国際学会講演（査読あり）

1. Ridai, T. and Kikai, H., History of computational fluid dynamics, [22nd Tokyo University of Science Conference](#) (TUSC22), Tokyo (Japan), 23–26th Sep. (20xx), Talk xx, 5 pages.

### 国際学会講演（査読なし）

1. Ridai, T. and Kikai, H., History of computational fluid dynamics, [22nd Tokyo University of Science Conference](#) (TUSC22), Tokyo (Japan), 23–26th Sep. (20xx), Talk xx, 5 pages.

## 国内学会講演（査読なし）

1. 理太郎, 機械花子, 数値流体力学の歴史, 第 22 回東京理科大学学会, 東京, 9 月 23–26 日 (20xx), Talk xx, 5 pages.

## 付録 B

### スーパーコンピューターごとの性能比較

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.