

タイトル: L^AT_EX で文章を書く

著者: 19312289 小熊 隼

日付: 2022/2/14

1 はじめに

T_EX は、文書をきれいに書くための組版処理システムです。また、L^AT_EX は T_EX をもとに更に文章を書きやすくしたものです。文字の配置や数式・画像の参照などを柔軟に解決できるので論文やレポートを書くのに非常に便利ですが、環境設定や記法などに若干の取っ付きにくさがあります。

ここでは基本的な設定や記法について解説しています。これを読んで少しでも T_EX および L^AT_EX の便利さを理解して親しんでもらえればと思います。

なお T_EX にはいくつかの入手方法がありますが、ここでは最も一般的な TeX Live[1] をおすすめします。

目次

1	はじめに	1
2	環境設定	2
2.1	T _E X の入手	2
2.2	L ^A T _E X エディタ	2
2.3	ブラウザ上での L ^A T _E X	2
3	L ^A T _E X の書き方	2
3.1	コマンド	3
3.2	プリアンプル	3
4	文章の構成	4
4.1	本文 (document)	4
4.2	章立て (section, subsection, subsubsection)	4
4.3	列挙	4
4.4	コメントアウト	4
5	数式・図表	5
5.1	数式の入力	5
5.2	図の挿入	5
5.3	表の作成	5
6	クロスリファレンス	7
7	参考文献の書き方	7
8	他にも...	8

2 環境設定

2.1 TeX の入手

前述のサイト [1] などから TeX をダウンロード、インストールする。ポピュラーな手段であれば自動的に LaTeX や主要なパッケージも同時にインストールされる。場合によっては環境変数などに追記が必要かも。(詳しくは TeX Wiki[2] に譲る)

2.2 LaTeX エディタ

TeX Live には TeXworks Editor というエディタが同梱されているが、補完やコンパイルの面であまり使い勝手がよくない。Visual Studio Code(以下、vscode) が、他の言語にも使えて拡張性もあるのでおすすめ。

vscode をインストール後、拡張機能から 'LaTeX Workshop' を追加、設定ファイルとして 'settings.json' を編集する。ここがやや面倒なので、自分の使っている設定ファイルを同梱する。

2.3 ブラウザ上での LaTeX

上記のような環境構築が面倒という人は、ブラウザ上で完結する各種サービスを用いるのも良いかもしれない。例えば、CloudLaTeX[3] や Overleaf[4] など。デフォルトで自動コンパイルが有効になっており、また基本的なパッケージは利用可能。

サインアップの必要はあるが、ファイルのクラウド上での編集や共有なども容易なので、ローカルでの作業が不要な場合には一考の余地があるかもしれない。

3 LaTeX の書き方

ここからは、実際に LaTeX で文を書くのに必要な基本事項を示す。

流れとしては、

1. .tex ファイルの編集
2. 保存
3. コンパイル x 2

を繰り返す。TeXworks Editor ではウィンドウ左上のタイプセットを 2 回押すことで実行、vscode では (設定によっては)Ctrl+S のみで OK。



図 1 TeX Live での 'タイプセット'

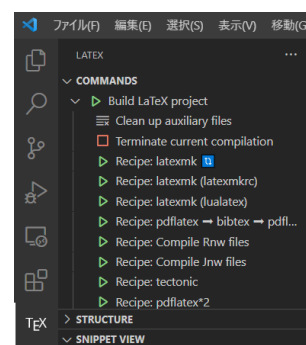


図 2 vscode でコマンドを選択する場合

ここで選択できるプログラムにはいくつかの種類があるが、日本語で PDF を出力したい場合は基本的には 'ptex2pdf' を選択する。なお、各種プログラムのために必要な設定を ".latexmkrc" ファイルに記載して同梱し

た。このファイルを”C:/Users/ユーザ名/”下に配置することで各プログラムの挙動を制御できる (Mac, Linux はホーム”~”下)。

3.1 コマンド

TeX では、文字や図表の制御にバックスラッシュ’\’で始まるコマンドを使う。

代表的なものとしては、ファイルの先頭に書く’\documentclass’や’\usepackage’、特殊な環境の範囲を示す’\begin’/’\end’などがある。

また、多くのコマンドは引数を取り、波括弧 {} 内に主要な引数を、角括弧 [] 内にオプションを記述する。
(例. ’\documentclass[a4j, dvipdfmx]{jsarticle}’)

二重バックスラッシュ’\’は強制的な改行を意味するほか、いくつかの特殊文字は命令の一部になっていたりするので、コマンドによってのみ出力できる。バックスラッシュ自身は’\textbackslash’で出力できる。なお、TeX 上に空行 (改行 2 回) を挿入することでも改行でき、改行 1 回は出力では無視される。

3.2 プリアンブル

’\documentclass’や’\usepackage’など、本文の前に書く部分をプリアンブルという。’\pagestyle’などを使えば、体裁を変えた文書にできる。

documentclass は文書全体の書式を決定し、usepackage は便利な拡張パッケージを有効化する (例. amsmath, graphicx)。maketitle のタイトル、著者、日付の設定もここで行なう。

また、よく使う設定はファイル (例. preamble.tex) にひとまとめにしておいて ’\input{preamble.tex}’などで読み込めば、毎回書いたりコピペしたりしなくてよい。

4 文章の構成

4.1 本文 (document)

文章の本体は、`\begin{document}` と `\end{document}` に囲まれた部分からなる。document 内を空にしたままコンパイルするとエラーになるので注意。

4.2 章立て (section, subsection, subsubsection)

章立ては `\section` のようにして行なう。同様に、subsection、subsubsection と続く。各コマンドのカッコ内には、それぞれのタイトルを入れる。

4.2.1 なお subsubsection はこのような感じ

4.3 列挙

いくつかのことがらを列挙したい場合は、

- `'itemize'` や
- `'enumerate'`

を用いる。itemize は順序のない列挙を、enumerate は番号付きの列挙に用いられ、いずれも `'\item` 列挙する内容' と書く。

列挙の例

```
\begin{enumerate}
  \item ham
  \item egg
  \item spam
\end{enumerate}
```

1. ham
2. egg
3. spam

4.4 コメントアウト

特定の範囲をコメントアウトしたい場合、`%` を書くことで、これより右の文字はすべてコメントとして扱われる。

また、複数行のコメントは `usepackage` で `comment` を有効化した上で `begin/end {comment}` で囲む。パッケージを読み込むのが面倒な場合は `'\if0'` または `'\iffalse'` と `'\fi'` で囲むことでも一応コメントアウトできる(どちらかと言うとシーケンス制御でスキップしているような感じ。途中で空行を挟むと動作しないらしいので、無難に `comment` や `%` を使った方が安全)。

5 数式・図表

5.1 数式の入力

'\\begin/end{equation}' で、数式を表示できる。各種記号などを入力する際には、amsmath および amsfont パッケージを用いる。

数式の例

```
\\begin{equation}
  \\sigma = \\frac{1}{1+e^{-ax}}
\\end{equation}
```

$$\sigma = \frac{1}{1 + e^{-ax}} \quad (1)$$

また、\$で挟んだ部分を $y = fx$ のように簡易的に数式モードで表示できる。

数式に関しては膨大な記法が存在するのでここでは詳細な記述は避ける。

5.2 図の挿入

文中に図を表示したいときは、'figure' を用いる。なお、前提として graphicx パッケージを有効化しており、また documentclass または graphicx のオプションに dvipdfmx を記述しておく必要がある。

図の挿入の例

```
\\begin{figure}[ht]
  \\begin{center}
    \\begin{tabular}{c}
      \\includegraphics[scale=0.23]{fig/sigmoid.eps}
    \\end{tabular}
    \\caption{シグモイド関数}
  \\end{center}
\\end{figure}
```

オプション'ht(=here, top)' は図の挿入位置を、'scale' は元画像からの縮尺をそれぞれ意味する。相対的な scale の代わりに高さを指定する height や幅を指定する width でも大きさを変えられる。また、ファイル名の拡張子はまぎれの無い場合省略可能であり、eps、pdf に加え png、jpg など一通りの形式はそのまま挿入できる。

さらに center によって中央揃えに、caption によって表題を表示できる。tabular は複数の図を並べる際に使えるが、単一の場合は省略してよい。

5.3 表の作成

表の作成には'table'を使用する。

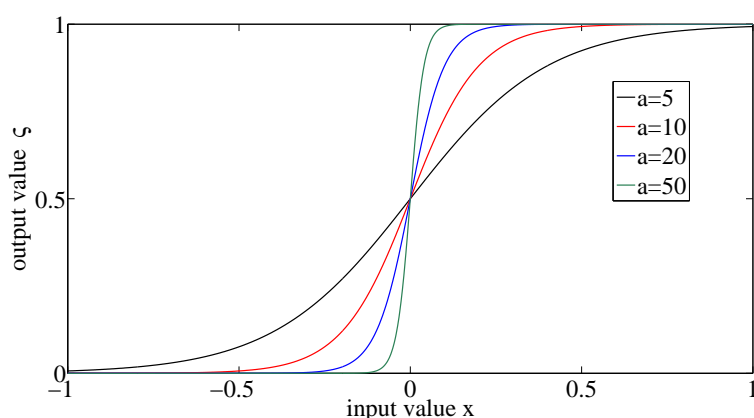


図3 シグモイド関数

表の例

```

\\begin{table}[ht]
  \\begin{center}
    \\caption{CPU Time [sec]}
    \\begin{tabular}{cccc} \\toprule
      Image size & V-RSSR & Conventional & Proposed \\
      500$\\times$300 & 0.17 & 4.08 & 0.33 \\
      1000$\\times$600 & 0.40 & 15.9 & 1.06 \\
      2000$\\times$1200 & 1.49 & 62.9 & 3.96 \\
    \\bottomrule
    \\end{tabular}
  \\end{center}
\\end{table}

```

表1 CPU Time [sec]

Image size	V-RSSR	Conventional	Proposed
500×300	0.17	4.08	0.33
1000×600	0.40	15.9	1.06
2000×1200	1.49	62.9	3.96

`tabular`によって表を区切る。`{cccc}`は行数と各行の文字詰め(ここでは'c'enterで中央揃え)を表し、各文字の間に縦棒'|'を入れることで縦罫線を引ける。また、'\\hline'で横の罫線を、`toprule`と`bottomrule`で外枠を描画できる。行の区切りは&で、列の区切りは改行文字で表す。

見ての通り手作業で打ち込むのはなかなか骨が折れるので、WEB上の変換サイトを使うか自分で変換用

スクリプトを書く、あるいはプログラムの出力を表にしたい場合は、最初から LaTeX 用のフォーマットで出力出来るようにしておくとい。また別の手段としては、excel などのソフトで作った表を図として保存して、前述の通り表示するやり方がある。PDF で閲覧したときにドラッグやコピーができないが、最終的に紙媒体などにする場合には違いがないのでこちらの方法でも問題ない。

6 クロスリファレンス

長い文章を書いているうち、数式番号や図番号などが煩雑になってくることがある。また、構成の都合で順番を入れ替えたり挿入・削除が行なわれると文章全体に修正が及ぶこともある。

LaTeX ではクロスリファレンスと呼ばれる機能を使って、各種番号を自動的に管理できる。

具体的な方法としては、まず参照したい数式や図表などに目印となるラベル `\label{ラベル名}` を付ける。次に、参照先で `\ref{参照したいラベル名}` の形でラベルを呼び出す。これで、数式・図・表などごとに管理された番号を特段意識せずに参照できる。

クロスリファレンス

```
\begin{equation} \label{eq:ohm}
V = R \cdot I
\end{equation}式

(\ref{eq:ohm}では)...
```

$$V = R \cdot I \quad (2)$$

式 (2) では...

注意点として、3 章で述べたように、クロスリファレンスを正常に表示するには 2 回のコンパイルが必要になる。また、数式や図表などで同じようなラベル付けを行なうと、名前の衝突などが起こる可能性がある。そのため、fig:や eq:, tab:など種類ごとにプレフィックス (接頭辞) を付けるとラベルを管理しやすい。

7 参考文献の書き方

参考文献の羅列は、文末に thebibliography を用いて bibitem を列記する形で行われる、と紹介されることが多いが、ここではより管理や列記が簡単な bibtex について解説する。

bibtex は文献管理のための tex 拡張で、多くの電子的な論文管理サイトでは bibtex 形式での引用がサポートされている。

まず、環境変数 BIBINPUTS を追加し、適当なディレクトリを bibtex のルートディレクトリとして設定する。続いて、設定したディレクトリに”ファイル名.bib”ファイルを作成する。ここに、bibtex で指定されたフォーマットで文献リストを書き込んでいく。先述の通り、IEEE や arXiv など多くのサイトでは”cite this”のリンクから bibtex 形式をコピーしたりダウンロードしたりできるので、これらを当該ディレクトリに配置したり追記したりしていけばよいだろう。

bib ファイルの記述例

```
@misc{texlive, % ← これがラベル名
  title="{TeX Live}",
  note="\url{https://tug.org/texlive/}"
}
```

そうして BIBINPUTS 下に参考文献のリストが出来たら、`TeX` ソース内に`\bibliographystyle` で引用スタイルを指定し (例. `junsrt`: 日本語、引用順)、`\bibliography{ファイル名(.bib)}` でソースとなる `.bib` ファイルを指定する。後は、文中で引用したい箇所に`\cite{bib ファイル内でのラベル名}` を書けばよい。

各々の記述が終わったら、まず 1 回 `ptex2pdf` に通す。その後、`pbibtex` を 1 度実行して、さらに `ptex2pdf` を 2 回実行することで、始めて参考文献リストと引用箇所の番号が揃う。つまり、`ptex2pdf` → `pbibtex` → `ptex2pdf` x 2 の計 4 回のプログラム実行が必要になる。ただし、`pbibtex` の実行は基本的には参考文献に更新があったときのみでよく、基本的には `ptex2pdf` x 2 でよい。

このとき、スタイルが `junsrt` であれば文中での登場順に文献リストが並び、また引用していない文献は、`bib` ファイル内にあってもリストには表示されない。

8 他にも...

ここで紹介した以外にも、様々なことができる。例えば

- 段組みを変える、部分的に変えたりも
- プログラムを外部から取り込んできれいに表示する
- PDF ファイルを取り込んで上から文字を書く
- 長くなった `TeX` ファイルを分割、コンパイルして 1 つの PDF にする

基本に慣れた後に不便を感じるものがあったら、調べてどんどん自分で便利にしていこう。

参考文献

- [1] TeX Live. <https://tug.org/texlive/>.
- [2] TeX Wiki. <https://texwiki.texjp.org/?TeX%E5%85%A5%E6%89%8B%E6%B3%95>.
- [3] Cloud latex. <https://cloudlatex.io/ja>.
- [4] Overleaf. <https://www.overleaf.com/>.