数学論文の書き方 第 1 回

Karel Svadlenka（オリジナルバージョン：荒井 迅・宍倉 光広）

2018 年 5 月 11 日

目次

1 はじめに

1.1 この講義について

■この講義の目的

本講義は修士課程の学生を対象に，論文の作成や学会プレゼンテーションの**技術的な側面**を解説するものです．

背景

- 修士論文提出間際になって TeX が使えなくてあせる学生が毎年続出している．
- 学会講演や学振特別研究員の面接において数学の学生のプレゼンテーションが下手で損をしている．

単位認定はレポートによって行ないます．（〆切 *6* 月 *1* 日）

■講義資料とインストール方法

講義の資料

講義の資料（このスライドの内容などのファイル）は，この講義の PandA のサイトで閲覧・ダウンロードできる．

TeX のインストール

- **自分のパソコンを使う** 次のようなサイト（とそこからリンクされているサイト）で自分に一番合う環境を選んで，TEX をインストールすることができる： https://texwiki.texjp.org www.comp.tmu.ac.jp/tsakai/lectures/intro_tex.html
- **大学の計算機室を使う** あとで，使い方を簡単に説明する．

1.2 参考文献

■論文の「中身」についての参考文献

木下是雄 「理科系の作文技術」 中央公論社 ISBN4121006240
定番中の定番．「自分は論理的な文章を書けている」と自負する人も読めば必ず得るところがあるでしょう．

野水克己 「数学のための英語案内」 サイエンス社 ISBN4781906923
Kobayashi-Nomizu の野水先生．すぐに使える例文が豊富．修論を英語で書く人は役に立つでしょう．

Strunk and White, *"The Elemtents of Style"*, Allyn and Bacon, ISBN020530902X
簡潔でしっかりした英語を書くための本．生協で平積みしてます．薄くて安くて一生役に立ちます．

■LATEX に関する参考文献

奥村晴彦 「*LATEX 2ε* 美文書作成入門」 技術評論社 ISBN：978-4774160450
LATEX のインストールから，書籍の作成に必要な知識まで解説してある便利な本．Windows, Mac OSX 用の LATEX インストール DVD も付属．

小田忠雄 「数学の常識・非常識—由緒正しい *TEX* 入力法」
http://www.math.tohoku.ac.jp/tmj/oda_tex.pdf
LATEX の使用法や数学の英語表現解説に関する間違いを集めたもの．Tohoku Math. Journal の編集に携わる上で著者が出会った実例に基づいています．一度は目を通すべし．

2 論文の内容について

修士論文の内容については，指導教員の方とよく相談して下さい．いろいろな論文を読んで，その構成・Introduction の説明の仕方・様々な定義や既知の事実の導入・文章表現などを参考にして下さい．

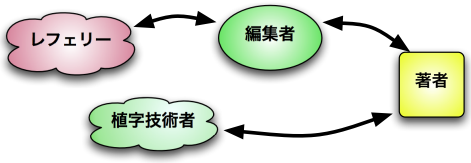
標準的な論文の構成

- タイトル・著者名
- Abstract: 論文の要約
- Introduction・序 この分野の目標，過去の研究，この論文の研究の意義，主結果の概要，謝辞など
- Preliminaries・準備: 必要な定義，事実など
- 本文
- References・参考文献

3 LATEX とは

■論文が出版されるまでの流れ

***TEX* 誕生以前**



1. **著者**が内容を思いつく
2. **著者**は原稿を手書きもしくはタイプして編集者に送る
3. **編集者**は referee に意見を求めて論文の採否を決定する
4. **著者**は referee の意見に従い内容の修正をする
5. **植字技術者**は原稿を元に植字して版を作る
6. **著者**は送られて来た試し刷りを校正し送り返す
7. **植字技術者**は誤植を訂正する
8. 論文が Journal に掲載される

■TEX がなぜ必要になったか

問題点

- 手書きの原稿は読みにくい
- 植字技術者は数学の素人の場合が多く，誤植が多発
- 高い技術を持つ植字技術者がだんだん減って来た

- ここで Don こと Donald Ervin Knuth 先生が登場.

- “The Art of Computer Programming” を書いていた先生は組版の汚なさに立腹し出版社と喧嘩.
- 本の執筆は中止してコンピュータ組版ソフトの開発を始めてしまった．TEX の誕生である.

■TEX から LATEX そして pLATEX へ

TEX は美しい組版と便利さで世界中に広まったが，現在我々が「TEX を使う」という場合には TEX そのものではなく Leslie Lamport が TEX を元に開発した LATEX を用いる場合が多い.

TEX や LATEX では日本語が使えなかったが，（株）アスキーが日本語化したものを配布していた．これを pLATEX と呼ぶ．現在は，日本語 TEX 開発コミュニティが開発を受け継いでいる.

TEX はなんて読むの？
本来はギリシア読みして「テッハ」．英語圏では「テック」日本では「テフ」と読む人が多い.
LATEX は英語圏では「ラテック」または「レイテック」，日本では「ラテフ」と読む人が多い.

■TEX は何であって何でないか

TEX は

- 組版（くみはん）ソフトである．
- Word のようなワープロでは**ない**．
- マークアップ言語処理系である．

マークアップ言語とは
コンピュータ言語の一種であり，文章の中に「文章の構造」を指定する「タグ」を埋め込んだものである．

HTML　Web ページの記述に用いられている言語．
TEX/LATEX　いま講義してるこれ．
MathML　Web での利用を意識して開発されている数式記述言語．要するに Web ブラウザで綺麗な数式を表示しようという試み．

4　使ってみる

■始めの一歩
最も単純な LATEX プログラムは次のようなものです．

```
\documentclass{jsarticle}
\begin{document}
ここに何か書いてください．
\end{document}
```

注意

- 「ここに何か書いてください」以外は半角で入力する．
- 環境によっては \ は円記号 (¥) になっている．

コンパイル結果
ここに何か書いてください．

■LATEX 命令
さっきの文章で印刷されなかった部分は LATEX への指令です．LATEX への指令は通常 \ で始まります．

\documentclass{jsarticle}	**文書の種類を指定**
\begin{document}	**本文の始まりを指定**
ここに何か	
書いてください．	
\end{document}	**本文の終わりを指定**

注意

- 本文中の改行やスペースは基本的に無視されます．
- 「どこで改行すれば綺麗に見えるか」等の見栄えに関することは TEX が考えてくれます．
- 文章の見栄えに関することは全て TEX に任せ，人間は文章の構造と中身に集中すること．

■簡単な数式

LaTeX は $ で囲まれた部分を数式だと思って処理します.

```
\documentclass{jsarticle}
\begin{document}
たとえばこんな
$f_0 := x^2 + y^2$
数式.
\end{document}
```

コンパイル結果

たとえばこんな $f_0 := x^2 + y^2$ 数式.

- 上付きの添字は ^ で表現する
- 下付きの添字は _ で表現する

■ギリシャ文字・分数・別行立ての数式

$$ で囲まれた部分は独立した行を占める数式になります.（\[と\] で囲んでも同じ.）

```
\documentclass{jsarticle}
\begin{document}
たとえばこんな
$$\Gamma = \frac{\alpha + \beta}{\sqrt{\gamma}}$$
数式.
\end{document}
```

コンパイル結果

たとえばこんな

$$\Gamma = \frac{\alpha + \beta}{\sqrt{\gamma}}$$

数式.

■略語由来の記号

sin, cos, tan

```
\documentclass{jsarticle}
\begin{document}
たとえば
$\exp(i \theta) = \cos \theta + i \sin \theta$.
\end{document}
```

コンパイル結果

たとえば $\exp(i\theta) = \cos\theta + i\sin\theta$.

注意
略語由来の記号は通常ローマン体で表記するためこのような命令が用意されています．`\sin` 命令を使わず単に `$sin x$`とすると $sin x$ のようになるので注意．

■和・積分など

```
ほげほげが$\lim_{n \to \infty} x_n$となるので
$$\lim_{n \to \infty} x_n -
\sum_{n = 0}^{\infty} s_n =
\int_{0}^{1} g(t) dt.$$
```

コンパイル結果
ほげほげが $\lim_{n \to \infty} x_n$ となるので

$$\lim_{n \to \infty} x_n - \sum_{n = 0}^{\infty} s_n = \int_{0}^{1} g(t) dt.$$

数式が独立しているかで添字の位置が違うことに注意．`$$ ... $$`の効果を文書のなかで出すには `$\displaystyle ...$`を使う．

■括弧
　数式に現れる括弧などを `\left` と `\right` を付けた組にすると括弧の大きさを自動で調整してくれます．

```
\documentclass{jsarticle}
\begin{document}
$$A=\left\{x \left| \left[\left(x-\frac12\right)^2
+1\right]^2 < 4 \right. \right\}$$
\end{document}
```

コンパイル結果

$$A=\left\{x \left| \left[\left(x-\frac12\right)^2 +1\right]^2 < 4 \right. \right\}$$

`{ }` は `\{ \}` で入力．ペアにしたくないもの（上の例の `\left|` など）は `\right.` と組み合わせる．

■改行・改段落
　文章を強制的に改行するには，`\\` を挿入します．通常の改行は 1 個ならば無視されますが，改行が 2 つ続いたらそれは段落の区切りと見なされます．`\par` でも段落の区切りとなります．改段落すると，環境によっては字下げ（インデント）されますが，それを抑制するには、`\noindent` を使います．

```
\documentclass{jsarticle}
\begin{document}
文章をここ\\ で改行します.\par ここから新しい段落です.
\end{document}
```

コンパイル結果
文章をここ で改行します．
　ここから新しい段落です．

■パッケージの利用

amssymb

LaTeX コマンドの中には，特殊なパッケージを使うと宣言してあげないと使えないものもあります．

```
\documentclass{jsarticle}
\usepackage{amssymb}
\begin{document}
たとえばこんな
$H^2(M) = \mathbb{Z} \oplus \mathbb{Z}_2$数式.
\end{document}
```

コンパイル結果

たとえばこんな $H^2(M) = \mathbb{Z} \oplus \mathbb{Z}_2$ 数式.

■行列

amsmath

行列を書くためには amsmath パッケージが便利です．

```
\documentclass{jsarticle}
\usepackage{amsmath}
\begin{document}
たとえばこんな
$\begin{pmatrix} 1 & 3  \\ 2 & 4 \end{pmatrix}$,
$\begin{bmatrix} 1 & 3  \\ 2 & 4 \end{bmatrix}$,
$\begin{vmatrix} 1 & 3  \\ 2 & 4 \end{vmatrix}$
行列たち.
\end{document}
```

コンパイル結果

たとえばこんな $\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$, $\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$, $\begin{vmatrix} 1 & 3 \\ 2 & 4 \end{vmatrix}$ 行列たち.

■論文ぽくしてみよう

著者名や文章のタイトル

著者名や文章のタイトルは「文章の構造」に関する情報なので，LaTeX に教えてあげれば勝手に整形して表示してくれます．

```
\documentclass{jsarticle}
\usepackage{amssymb}
\title{ドラゴンボール$\mathbb{Z}$}
\author{鳥山　明}
\begin{document}
\maketitle
```

なんやかんやあって孫悟空は地球を救った.
```
\end{document}
```

■$DB\mathbb{Z}$

コンパイル結果

ドラゴンボール $\mathbb{Z}$

鳥山 明

2018 年 5 月 11 日

なんやかんやあって孫悟空は地球を救った.

- タイトルに数式を使うのはなるべく避けたほうがよい
- 日付はコンパイルした日が自動的に入る．自分で指定する場合は `\date{1975 年 7 月 22 日}` などとする．

■コマンドの定義

よく使う記号などは`\newcommand`で新しい記号を定義すると便利.

コマンド定義の例

```
\documentclass{jsarticle}
\usepackage{amssymb}
\newcommand{\R}{\mathbb{R}}
\newcommand{\myfrac}[1]{\frac{1}{a+{#1}}}
\begin{document}
関数 $f:\R^2 \to \R$ を $f(x,y)=x+\myfrac{y}$ で定義する。
\end{document}
```

コンパイル結果

関数 $f:\mathbb{R}^2 \to \mathbb{R}$ を $f(x,y)=x+\frac{1}{a+y}$ で定義する。

■その他

句読点

{、}{。}　句点＆読点（全角）
{，}{．}　コンマ＆ピリオド（全角）
{,}　{.}　 コンマ＆ピリオド（半角）

日本語で数学の文章を書く場合，数式内に表われるコンマとの調和を考えてコンマとピリオドを用いるのが一般的です．

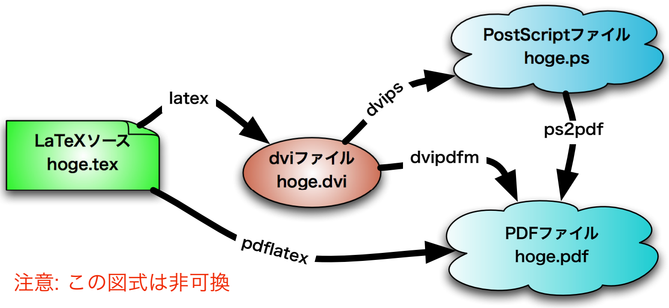
ピリオド＆コンマ

英語の文章では，ピリオドやコンマ，セミコロンといった役物の前にはスペースを入れないこと．

5　LaTeX の仕組み

5.1　処理の流れ

■LaTeX の処理の流れ



LaTeX ソース　さっき書いたやつ
dvi ファイル　TeX が生成する中間ファイル
PostScript ファイル　主に印刷用の形式
PDF ファイル　画面表示/印刷用の形式

■コマンドラインで LaTeX を使ってみよう

LaTeX の処理を理解するにはコマンドラインから使ってみるのが一番です．教室の UNIX にログインして次の命令を実行してみること．

emacs hoge.tex　　（*LaTeX* **ソースを編集**）
platex hoge.tex　　（*hoge.tex* **から** *hoge.dvi* **を生成**）
dvipdfmx hoge.dvi　　（*hoge.dvi* **から** *hoge.pdf* **を生成**）
atril hoge.pdf　　（*hoge.pdf* **の内容を表示**）

注意

- emacs でなくても好きなエディタを用いればよい
- 本文が英語の場合は platex でなく latex でもよい
- 日本語の文章を pdflatex で処理するには別パッケージの読み込みが必要

5.2　エラー対処

■コンパイルエラー

エラーメッセージの例

```
This is pTeX, Version 3.141592-p3.1.10 (utf8) (Web2C 7.5.4)
(./240.tex
pLaTeX2e <2006/01/04>+0 (based on LaTeX2e <2005/12/01> patch level 0)
(/usr/local/teTeX/share/texmf/ptex/platex/jsclasses/jsarticle.cls
Document Class: jsarticle 2006/11/01 okumura
)
! Undefined control sequence.
<recently read> \begn
```

```
l.2 \begn
         {document}

! LaTeX Error: Missing \begin{document}.
```

- 2 行目の`\begn` という命令は知らないよ
- `\begin{document}` がどこにもないよ

ということをこのエラーメッセージは訴えています.

■よくあるエラー

- 命令名が間違っている
- `{` と `}` が一対一に対応しない
- `\begin{hoge}` と `\end{hoge}` が一対一に対応しない
- 数式の終わりを示す `$` がない
  - `! Missing $ inserted....?`などと出たらこれを疑う
- `\left` で始めた括弧などの`\right` がない
- 半角文字で書くべきコマンドを全角で書いた
- パッケージを使う宣言を忘れた
- 使うと宣言したパッケージがインストールされていない
- `$f_a_c$`などと添字が二重になっている
  - 二重添字 f_{a_c} は正しくは `$f_{a_c}$`

■コメントアウト

LaTeX は % を見つけると，その行の % 以降を無視します.

```
\documentclass{jsarticle}
\begin{document}
% ほげほげ
ここに何か書いてください. % ふがふが
\end{document}
```

コンパイル結果
ここに何か書いてください.

コメントアウトを利用して，エラーの発生箇所を突きとめるのがエラー対策の第一歩になります.

■日本語の文字コード

文字コードの違いもトラブルの原因となります.

主な日本語の文字コード

ISO-2022-JP　日本語電子メールの標準規格
EUC-JP　昔の UNIX 上でもっとも広く利用されていた
Shift_JIS　Windows/Macintosh 上でよく利用されている

UTF-8　Unicode の代表的な符号化方式

教室の Linux 上の pLaTeX では

`platex -kanji={utf8, euc, sjis, jis}`

のように文字コードを指定できる

- `utf8, euc, sjis, jis` のうち，どれかを選ぶ
- 標準（何も指定しないとき）は `utf8`

■nkf

文字コード推定
あるファイルの文字コードが知りたい場合も nkf コマンドが使えます.

`nkf --guess hoge.tex`

文字コード変換
UNIX 上で文字コードを変換する場合は nkf コマンドを使うのが便利です.

`nkf -e hoge.tex > hoge_euc.tex`

により `hoge.tex` との内容を EUC-JP に変換したものが `hoge_euc.tex` というファイルに保存されます.

6　インストール

■数学教室の Linux の場合
既に必要なものはインストールされています.

emacs, vi　エディタ

latex, platex　LaTeX コンパイラ（英語/日本語）
tex, ptex　plain TeX コンパイラ（英語/日本語）
pdflatex　PDF を直接生成する latex （英語）
luatex　豊かなフォントが利用できる TeX（英語/日本語）

dvips　dvi から PostScript を生成
dvipdfmx　dvi から PDF を生成

atril　PDF ファイルを表示

■数学教室の Linux の場合

統合環境

TeXworks　簡単な初期設定が必要：計算機室の張り紙を参照すること

- platex2pdf-nkf というプログラムが用意してある（文字コードを推定して，dvi ファイルと pdf ファイルを順番に生成する）
- 日本語の設定は難しい

■情報環境機構が管理している計算機室の場合

理学研究科 6 号館 210 号室など

統合環境

- プログラムのなかにある EasyTeX または TeXworks を起動
- そのままテキストエディターにソースコードを入力
- 上のメニューにあるボタンを押すだけでコンパイルできる

日本語は問題なく使える

■インストール

- TeX / LaTeX はフリーソフトウェアである．
- 自分でソースコードからコンパイルすることもできる．

情報源

インストールを含め，TeX 関係で困ったら TeX Wiki

`https://texwiki.texjp.org`

を調べること．

■TeX / LaTeX のインストール

Internet からダウンロード

`https://texwiki.texjp.org/?Mac`
`https://texwiki.texjp.org/?W32TeX`
`www.comp.tmu.ac.jp/tsakai/lectures/intro_tex.html`
を参照すれば，Internet からダウンロードしてインストールする方法が書いてある．

CD からインストール

奥村晴彦 **「LaTeX 2ε 美文書作成入門」** 付属 DVD からインストールできる

数学論文の書き方 第 2 回

Karel Svadlenka （オリジナルバージョン：荒井 迅・宍倉 光広）

2018 年 5 月 18 日

目次

1 文章の構造

1.1 文章の種類

■amsart,...

LaTeX では文章の種類を \documentclass{} で指定する.

article 短かめの文章
jarticle article の日本語版
jsarticle article の新しい日本語版
book 長めの文章のためのスタイル
jbook book の日本語版
jsbook book の新しい日本語版
amsart AMS で採用される標準的な論文のスタイル

■クラスファイル

タイトルはどのくらいの大きさで出力するか，ページ番号をどこに出すかといった文章の種類に依存する設定は「クラスファイル」と呼ばれるファイル（拡張子 .cls）に入っている．

各ジャーナルが独自に開発したクラスファイル（例えば京大ジャーナル用のクラスファイルなど）は LaTeX の標準的配布には入っていません．ジャーナルの指定する場所からダウンロードするか，article などの標準的なクラスで作成しておいて，整形は出版社に任せます．

1.2　文章の論理構造

■文章の論理構造

chapter, section, subsection,...
典型的な論文に含まれる要素

タイトル等　`\maketitle` で作る
abstract　`\begin{abstract}`から`\end{abstract}`まで
introduction　section 1 をイントロにあてることが多い
sections　`\section{`**セクション名**`}`
subsections　`\subsection{`**サブセクション名**`}`
謝辞　位置は人により様々
本文
文献リスト　作り方は後述

本文に含まれる要素

- 定義，定理
- 証明
- 図，表

1.3　セクション・定理・式などの番号付け

■セクション・定理・式などの番号付け

LaTeX はセクション・定理・式などに自動的に番号を割り振ってくれる．ラベルを付けてそれを引用することにより，順番を変えたり，途中に新しいものを挿入したりしても，引用する番号がずれなくなる．（定理・式については後で説明）

`\section{`**最初のセクション**`}\label{firstsec}` **最初の本文.**
`\section{`**次のセクション**`}\label{secondsec}` **次の本文.**
セクション `\ref{firstsec}` **で証明したように. . .**

コンパイル結果

1 最初のセクション

最初の本文.

2 次のセクション

次の本文．セクション 1 で証明したように...

1.4　定理環境

■amsmath

amsthm パッケージは，定理や定義に適したのフォントの指定や，定理番号の管理を自動で行ってくれます．

```
\documentclass{jsarticle}
\usepackage{amsthm}
\newtheorem{dfn}{定義}
\newtheorem{thm}{定理}
\begin{document}
\begin{dfn}
いい感じの定義.
\end{dfn}
\begin{thm}
何かすごい定理.
\end{thm}
\end{document}
```

■

コンパイル結果

定義 1. いい感じの定義．　**定理** 1. 何かすごい定理.

- \newtheorem{**環境名**}{**実際の表記**}で定理環境を定義
- \begin{**環境名**}...\end{**環境名**}で環境を呼び出す

番号は環境ごとに独立に付けられます．番号を共通にしたい場合，例えば「定義」「定理」「補題」で共通の番号を使う場合は

```
\newtheorem{thm}{定理}
\newtheorem{dfn}[thm]{定義}
\newtheorem{lem}[thm]{補題}
```

などとします．

■定理番号の管理

- \label{**ラベル名**}でラベルを付ける
- \ref{**ラベル名**}とすると番号が表示される

```
\newtheorem{thm}{定理}
\newtheorem{lem}[thm]{補題}
\begin{document}
\begin{lem}
  \label{poincare}
```

ポアンカレの補題.
\end{lem}
\begin{thm}
\label{euler}
オイラーの定理.
\end{thm}
補題\ref{poincare}と定理\ref{euler}により云々.

コンパイル結果

補題 1. ポアンカレの補題. **定理** 2. オイラーの定理.

補題 1 と定理 2 により云々.

注意

- 定理の番号は自分で手打ちしてはいけない.
- 最初のコンパイル時には番号が??になるが，もう一度コンパイルすることにより正しい番号が入る.
- セクション番号・式番号・図番号等にも\label{}と\ref{}が使える．式番号でカッコ付きにするには，\eqref{} を使う.

■証明

証明を書く場合は proof 環境を使います.

\begin{thm}
何かすごい定理.
\end{thm}
\begin{proof}
自明.
\end{proof}

コンパイル結果

定理 1. 何かすごい定理. *Proof.* 自明. □

2 文献リストの作成

■文献リストの重要性

学術論文において，引用文献のリストはその論文自体の内容と同じくらい重要な要素です．決しておろそかにしないこと.

注意

- 引用する論文の雑誌名，ページ番号などは必ず論文の現物で確認する．他人の論文の文献リストを丸写しするのは厳禁.
- 文献リストに必要な情報の種類や文献を並べる順番，また雑誌名の省略方法などは掲載する雑誌により異なります.

■bibliography 環境

- thebibliography 環境内に各文献の情報とラベルを列挙
- \cite{**ラベル名**}とするとその文献の番号が表示される

```
\documentclass{jsarticle}
\begin{document}
Caffarelli の結果\cite{Caf81}により云々.

\begin{thebibliography}{99}
  \bibitem{Caf81} L.A. Caffarelli, H.W. Alt,
    {\it Existence and regularity for a minimum problem
         with free boundary},
    Journal f\"ur die reine und angewandte Mathematik
    {\bf 325} (1981), pp. 105--144.
\end{thebibliography}
\end{document}
```

コンパイル結果

Caffarelli の結果 [1] により云々.

参考文献

[1] L.A. Caffarelli, H.W. Alt, *Existence and regularity for a minimum problem with free boundary*, Journal für die reine und angewandte Mathematik **325** (1981), pp. 105–144.

注意

\begin{thebibliography}{99} の 99 は，文献番号を表示するのに必要な桁数を決めるための指定です．引用文献が 99 本以下なら 99 のままで構いません.

■BiBTEX を使う方法

参考文献のデータベース（bib ファイル）を作り，それを TEX ファイルで読み込むことで，引用する文献のみ表示される.

bib ファイルの形式

```
@Article{Caf81,
        title = "Existence and regularity for a minimum problem with free boundary",
        author = "L.A. Caffarelli and H.W. Alt",
        journal = "Journal fur die reine und angewandte Mathematik",
        issue = 325,
        pages = "105-144",
        year = 1981,}
```

使い方

- 上記形式の文献データベースを bunken.bib として保存
- TEX ファイルの冒頭に `\bibliographystyle{jplain} \bibliography{bunken}`
- 本文中で\cite{Caf81} として文献を参照する
- コンパイル： `platex > jbibtex > platex > platex`

3　式の表示と式番号

■式と式番号

$$ で囲む代わりに，\begin{equation} と \end{equation} で囲むと式番号が付与される．\label{} でラベル（名前）をつけておけば，\eqref{} でその式番号を引用できる．

```
\begin{equation} \label{myeq1}
\sum_{n=1}^\infty \frac{1}{n^2} = \frac{\pi^2}{6}
\end{equation}
上の公式 \eqref{myeq1}を証明してみよう.
```

コンパイル結果

$$\sum_{n=1}^\infty \frac{1}{n^2} = \frac{\pi^2}{6} \tag{1}$$

上の公式 (1) を証明してみよう．

■式と式番号 2

後述の式環境 multline, gather, align でも式番号が付与される．\label{} と \eqref{} で式番号を引用できる．式番号を付けたくなければ * を付けて \begin{equation*} などとするか，\nonumber を挿入すれば番号はつかない．

```
\begin{equation*}
a^2+b^2=c^2
\end{equation*}
```

コンパイル結果

$$a^2 + b^2 = c^2$$

セクションごとに式番号を付けて 3 節の 2 番目の式を (3.2) などとしたい場合は，\begin{document} の前に，\numberwithin{equation}{section} と書く．

■複数行にわたる数式: align

\usepackage{amsmath} が必要．改行したいところに \\, そろえたいポイントに & を挿入．

```
\begin{align}
x^3+y^3 &= (x+y)(x^2-xy+y^2) \\
&= (x+y) \left( (x+y)^2-3xy \right)
\end{align}
```

コンパイル結果

$$\begin{align} x^3+y^3 &= (x+y)(x^2-xy+y^2) \tag{2} \\ &= (x+y) \left((x+y)^2-3xy \right) \tag{3} \end{align}$$

1 行あたり & を複数回使うと左揃え，右揃えが交互に変わる．

■**複数行にわたる数式**: split

`align` と同様であるが，`equation` 環境の中で使える `split`．改行したいところに \\, そろえたいポイントに & を挿入．

```
$$H(x) = \left\{
\begin{split}
&0 &\quad &(x<0) \\
&1 &\quad & (0 \le x)
\end{split} \right. $$
```

コンパイル結果

$$H(x) = \left\{ \begin{split} &0 &\quad &(x<0) \\ &1 &\quad & (0 \le x) \end{split} \right.$$

数式環境`$$`などのなかに`\begin{array}...\end{array}`を使っても同じ効果がある．

■**中央揃えの** gather，**長い 1 行を折り返す** multline

```
\begin{gather}
a_0 = 1, \ a_1=1 \nonumber\\
a_{n+1}=a_n + a_{n-1}
\end{gather}
\begin{multline}
a+b+c+d+e+f+g+h+i+j\\ +k+l+m+n+o+p+q+r+\dots+x+y+z
\end{multline}
```

コンパイル結果

$$a_0 = 1, \ a_1=1$$
$$a_{n+1}=a_n + a_{n-1} \tag{4}$$

$$a+b+c+d+e+f+g+h+i+j$$
$$+k+l+m+n+o+p+q+r+\dots+x+y+z \tag{5}$$

4　可換図式の作成

■amscd: **簡単な例**

```
$$
\begin{CD}
A     @>>> B     \\
@VVV       @VVV \\
C     @>>> D
\end{CD}
$$
```

コンパイル結果

$$
\begin{tikzcd}
A \arrow[r] \arrow[d] & B \arrow[d] \\
C \arrow[r] & D
\end{tikzcd}
$$

■amscd パッケージの基本

- プリアンブルには`\usepackage{amscd}`と書いておく
- 射は↑が`@AAA`，↓が`@VVV`，→が`@>>>`，←が`@<<<`
- 水平な射と対象を並べ，改行記号 `\\`を置く
- 鉛直な射はそれだけで独立した行を占める
- `@=` と `@|` で横と縦の長い等号
- 矢印の脇に数式を表示したいときは，`@>f>>`のように矢印記号の 3 つ組の間に数式を挟む
- 矢印のどちら側に数式が表示されるかは，数式を挟む位置で決まる

■amscd: 少し複雑な例

```
$$
\begin{CD}
       A    @>f>>  B   @>>g> C          \\
       @V{\phi}VV  @|        @VV{\psi}V \\
       C    @=     E   @>h>> F
\end{CD}
$$
```

コンパイル結果

$$
\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "\phi"'] & B \arrow[r, "g"'] \arrow[d, equal] & C \arrow[d, "\psi"] \\
C \arrow[r, equal] & E \arrow[r, "h"] & F
\end{tikzcd}
$$

■xypic: 簡単な例

より複雑な図式には xypic がよく利用される．プリアンブルには`\usepackage[all]{xy}`などと書いておく．

```
$$
\xymatrix{
A \ar[rr] &   & B \ar[dl] \\
          & C \ar[ul] &
}
$$
```

コンパイル結果

$$
\begin{tikzcd}
A \arrow[rr] & & B \arrow[dl] \\
& C \arrow[ul] &
\end{tikzcd}
$$

■xypic パッケージの基本

- xy*matrix* というくらいなので，対象は pmatrix のときのように &と\\で並べる
- ある対象をドメインとする射は，その対象の後に `\ar[ul]` という形で示される
- 射のコドメインは `[]` の中のアルファベットで示される（上下左右がそれぞれ u d l r）
- 射の名前を表示するときは`\ar[r]^f` もしくは`\ar[r]_f` のように添字風に表示する
- `\ar[r]|f` とすると矢印の途中に射の名前が表示される
- 矢印の種類は`\ar@{=>}`のように`\ar` の後に指定する

■少し意味不明な図式

```
$$\xymatrix{
A \ar@{=>}[rr]|f & & B \ar@{.>}[d]^g \\
C \ar@{=}[r] & D \ar[ul] \ar@{~>}[ur] & E \ar[l] \\
F \ar[u] \ar[r] \ar@{}[ur]|{\circlearrowleft}
& G \ar[u]
}$$
```

コンパイル結果

$$\begin{tikzcd}
A \arrow[rr, Rightarrow, "f" description] & & B \arrow[d, dotted, "g"] \\
C \arrow[r, equal] & D \arrow[ul] \arrow[ur, squiggly] & E \arrow[l] \\
F \arrow[u] \arrow[r] \arrow[ur, phantom, "\circlearrowleft"] & G \arrow[u]
\end{tikzcd}$$

■曲る矢印

```
$$\xymatrix{
  A \ar@(ur,ul)[rr]^f \ar[r] &
  B  \ar[r] &
  C \ar@(dl,dr)[l]^g
}$$
```

コンパイル結果

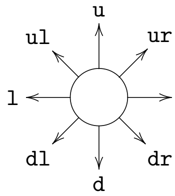
$$\begin{tikzcd}
A \arrow[r] \arrow[rr, bend left, "f"] & B \arrow[r] & C \arrow[l, bend left, "g"]
\end{tikzcd}$$

■曲げ方の基本

ドメインから出て行く方向とコドメインに入る方向を

`\ar@(`**出，入**`)`

という形で指定する．方向としては下の図の値が選べる．



曲げ過ぎ注意

あまり曲げ過ぎるとコンパイルが遅くなる．

5　図の取り込み

■graphicx

LaTeX 文章内に図や写真を取りこむには graphicx パッケージを用いる．

```
\documentclass{jsarticle}
\usepackage{graphicx}
\begin{document}
たとえば don の顔
\includegraphics{knuth.eps}
を取りこむ.
\end{document}
```

コンパイル結果

たとえば don の顔を取りこむ．

■Encapsulated PostScript

画像のフォーマット

LaTeX に取り込むには Encapsulated PostScript (EPS) 形式が最もトラブルが少なく，仕上りも良くなる．教室の UNIX 上ならば，例えば

```
convert hoge.gif hoge.eps
```

を実行すると `hoge.gif` の画像を `hoge.eps` に変換出来る．

図の作成

教室の Linux 上では tgif や xfig を使うのが手軽．

■figure 環境

図や画像は，figure 環境に入れると LaTeX が場所も自動で調整してくれるし，番号も管理してくれるので便利．

- `\begin{figure}`から`\end{figure}`までが figure 環境

- figure を表示する場所は LaTeX が自動で選んでくれる
- 場所に関する人間の希望もある程度は聞いてくれる
  - 希望する場所を`\begin{figure}[htb]` のように `[]` 内に優先順位の高いものから並べる

  h　この場所に表示

  t　ページの上部に表示

  b　ページの下部に表示

  p　図だけの独立したページを作成する
- 図のキャプションは`\caption{}`命令で与える
- 定理環境と同じように`\label{}`と`\ref{}`が使える

■figure 環境の使用例

```
\begin{figure}[tbh]
  \centering
  \includegraphics{knuth.eps}
  \caption{Don. Knuth}
  \label{don}
\end{figure}
```

コンパイル結果

図 1　Don. Knuth

■他の画像ファイルを挿入

jpg, png などの画像ファイルや pdf を配置することもできる.

Bounding box の指定
通常は bounding box を数値で指定するか：

```
\includegraphics[height=6cm, bb=0 0 300 400]{fig.jpg}
\includegraphics[scale=0.5, bb=0 0 300 400]{fig.jpg}
```

または，bb ファイルを作成し，画像ファイルと同じフォルダーに置く：
コマンドラインで　`extractbb fig.png`

6　論文投稿時の注意

■編集者へ原稿を送る

投稿したい雑誌が決まったら，その雑誌の投稿規定を熟読し，投稿の方法と，選べるならどの編集者に送るかを決定する.

主な投稿方法

郵送　船便にならないように注意.
email　添付し忘れないように注意.
Web form　サーバーが受け付けてくれたか確認すること.
arXiv **経由**　ミラーリングの時間差に注意.

注意

- 論文には必ずカバーレターをつけること.
- 雑誌によっては掲載料金を取るところもある.
- Open access の条件を確認すること.

7　プレゼンテーション

■プレゼンテーションの種類

- 黒板・白板
- OHP スライド
  - 手書き
  - コンピュータで作成
  - ライブ手書き
- PC プロジェクター
  - Keynote/PowerPoint プレゼンテーション
  - PDF ビューアを用いたプレゼンテーション

やってはいけないプレゼンテーション

- 論文をそのまま写影

■黒板・白板を用いたプレゼンテーション

メリット

- 壊れない確実さ
- 理解しやすいスピード
- 臨場感

デメリット

- 字が汚いと読めない
- 板書量が多いこと
- 計算データの提示が苦手
- 最近では会場に黒板・白板がない・少ない・小さい

注意点

- 事前に時間をはかることを忘れないように
- 黒板・白板に向いたままにならぬように

■OHP を用いたプレゼンテーション

スライド作成方法の種類

- 手書き
- 計算機で作成
- ライブ手書き

注意点

- 水性ペンで書いたりインクジェットで印刷したシートは湿度に弱いので注意
- OHP 用の油性ペンでホワイトボードに書き込まないこと
- 字の大きさが小くなり過ぎないように
- 会場の大きさをできれば事前にチェックすること

■コンピュータを用いたプレゼンテーション
メリット

- 綺麗，マルチメディア
- 再利用が容易

デメリット

- 速くなりがち
- トラブルが多い

注意点

- 電源アダプタ，VGA/HDMI アダプタなどを忘れない
- USB メモリにバックアップを保存しておく

最近では「beamer」という LaTeX のクラスを使う人が多い．（この資料もそう．）

■見やすい文字，色使い

- 文字はなるべく大きくする．
- 同じ大きさなら，一般的にサンセリフ系のフォントの方が見やすい．

日本人の場合，男性の 20 人に 1 人，女性の 400 人に 1 人は赤と緑の区別がつきにくいいわゆる「赤緑色覚異常」です．

- 黒板に赤チョークは使わない

マイク使用時の注意

- マイクと口元の距離を一定に保つように注意すること．
- マイクを付けたままスピーカーより前に出ないこと．

■Beamer

プレゼンテーション用に作られた LaTeX のクラス

PDF ファイルを作成し，Adobe Acrobat などで表示してプレゼンテーションを行う．

この講義のスライドも Beamer を用いて作成．

```
\documentclass{beamer}  % beamer を指定
\usetheme{Warsaw} % いろいろなスタイルが選べる
\title[Short title]{Full title of the Presentation}
\author[Short Name]{Full name of the author}
\institute[Short Inst]{Name of University}
\date{2018 年 5 月 18 日}
\begin{document}
```

\begin{frame} \maketitle \end{frame}

■Beamer (frame)

画面の表示単位がフレーム

`\begin{frame}` と `\end{frame}` の間にその画面に表示したい内容を書く．`\frametitle{}` によりそのフレームの題名を付ける．

\begin{frame} \frametitle{ フレームの題名 } フレームの内容 1 *pause* フレームの内容 2 \end{frame}

途中に \pause を挿入することにより，項目を順番に表示させることができる．

■Beamer (**箇条書き**)

箇条書き (enumerate など) でも \pause は使える．

```
\begin{enumerate}
\item enumerate を使った表示 \pause
\item 番号付きのリスト \pause
\item item ごとに表示できます.
\end{enumerate}
```

コンパイル結果

1. enumerate を使った表示
2. 番号付きのリスト
3. item ごとに表示できます．

注意：箇条書き (enumerate, itemize, description) は通常の (beamer 以外の)LaTeX でも使える．

■Beamer (block)

ここに表示されているような箱 (block) も使える．

```
\begin{block}{}
```

```
これはタイトル無しの箱
\end{block}
\pause
\begin{block}{箱のタイトル}
これはタイトル付きの箱
\end{block}
```

これはタイトル無しの箱

箱のタイトル
これはタイトル付きの箱

8　ポスター作成

■作成の方法

Adobe Illustrator を使う
そもそもポスターの作成のためのソフトウェアなので，大判ポスターの作成に必要な機能が揃っているが，数式は苦手.

Keynote 等の場合と同様に，LaTeX を用いて数式を作成し，PDF のカット＆ペーストで張り付ける.

院生控室のコンピュータで Illustrator を使用する必要がある場合，事務に申し出てください.

■LaTeX を用いる

a0poster
LaTeX で大判ポスターを作成する場合，a0poster というクラスを用いるのが便利.

使用例

```
\documentclass[portrait,a0]{a0poster}
\usepackage{multicol}
\begin{multicols}{3}
本文
\end{multicols}
```

上の例では multicol パッケージを使用することにより本文を 3 段組みにしている.

■A0 サイズの PDF を作成

hoge.tex というファイルから A0 サイズの PDF を作成するには以下のようなコマンドを実行する.

```
platex hoge.tex
dvips -o hoge.ps hoge.dvi
ps2pdf -sPAPERSIZE=a0 -r600x600 hoge.ps hoge.pdf
```

注意
A0 で作成したファイルを教室のプリンタに送ってもエラーで印刷できない．画面上で確認するか，確認段階では A4 で作成する.

■**出力**

A0 などの大きなサイズの紙に印刷するためには特殊なプリンターが必要になる．

数学教室の大型プリンタ

図書室の地下に大型のプリンタがあります．修士以上で研究集会発表などのためならば，利用することができます．PDF ファイルを USB メモリなどで持参し，図書事務室で問い合わせて下さい．

9　レポート

■**レポート**

課題

LaTeX を用いて論文を作成し，その PDF ファイルを提出すること．内容や使用言語は問わない．（数学的に意味のない内容でも構わない．）ページ数は 2〜3 ページ程度でよいが，

- 論文題名・著者名（学生番号を明記すること）・アブストラクト（概要）をもつ
- 本文にいくつかの定義，定理，補題などを入れ，その定理番号などを他で引用する
- 何種類かの数式を入れる．可換図式または行列を入れる
- 図（他のソフトウェアで作成または既存の図を引用でよい）を入れる
- 文献リストを作り，本文中で引用する

をみたし，完全な論文の体裁を持つようにせよ．図は，
`http://www.math.kyoto-u.ac.jp/files/latex/MathDeptEntrance.eps`
(数学教室玄関の写真) または京都大学マークを使ってもよい．

提出方法

作成した PDF ファイルの名前を「NNNNNNN 氏名.pdf」（ただし，NNNNNNN は学生番号）とし，それを，PandA の「数学論文の書き方」コースサイト内の「課題」において添付ファイルとして提出せよ．

PandA のサイトにアクセスするには，京大の「全学生共通ポータル」>「学習支援システム PandA」>「利用」を選び，右上の「ログイン」> ECS-ID とパスワードを入力する．PandA 内で，「マイワークスペース」>「メンバーシップ」を選択すると，既に KULASIS で履修登録していれば，この講義が参加しているサイトのリストに現れるはず．

提出期限

2018 年 6 月 1 日