

L^AT_EX 入門

秋山進一郎

筑波大学計算科学研究センター

2024 年 10 月 16 日

1 L^AT_EX による文書作成にあたって

まず、新しい文書を作成するごとにそれ専用のディレクトリを準備する。文書の名前はディレクトリ名で管理する。以下では、このディレクトリをメインディレクトリと呼ぶ。まずはメインディレクトリの中に.tex ファイルを作成する。文章に掲載する図は画像ファイルとしてメインディレクトリに格納する。図は全てベクターファイルで格納する。ベクターファイルを使う理由は、図の拡大、縮小時に解像度が低下しないためである。代表的なベクターファイルは.pdf, .eps などである。^{*1} 引用文献は BibTeX で管理することとし、.bib ファイルを作成する。この.bib ファイルもメインディレクトリに格納する。

ここでは、親となる.tex ファイルには main.tex と名前を付けることにする。例えば、Overleaf でもデフォルトの.tex ファイル名は main.tex である。画像ファイルは fig という名前のディレクトリを作ってそこに格納する。同様に、bib ファイルは bib という名前のディレクトリを作って格納する。例えば、卒業論文を作成する場合なら、senior_thesis という名前のメインディレクトリを作成し、その中に main.tex を作成する。そして、senior_thesis ディレクトリの中に fig ディレクトリと bib ディレクトリを作成する。実験や数値計算の結果をまとめた画像ファイルは全て fig ディレクトリに格納する。参考文献や引用文献の情報は全て.bib ファイルで管理し、bib ディレクトリに格納する。

なお、Overleaf で日本語入力ができるようにするためには、メインディレクトリに latexmkrc という名前のファイルを作成し、以下の内容を latexmkrc にコピーしておく必要がある。その後、左上の「Menu」から「Compiler」へ行き、「pdfLaTeX」を「LaTeX」

^{*1} ピクセルで構成されているラスターファイル (.jpeg, .png など) の使用は避ける。

に変更する.

ソースコード 1 latexmkrc

```
1    $latex = 'uplatex';
2    $bibtex = 'upbibtex';
3    $dvi2pdf = 'dvipdfmx %0 -o %D %S';
4    $makeindex = 'mindex -U %0 -o %D %S';
5    $pdf_mode = 3;
```

2 パッケージ

パッケージを指定することにより, 様々な数学表現, 物理学の記法, 書体などが L^AT_EX で使用可能になる. パッケージを指定するには, `\usepackage{ パッケージ名 }` をプリアンブル (Preamble) に書けば良い. プリアンブルとは, .tex ファイルの冒頭から `\begin{document}` までのことを言う. ソースコード 2 の 1 行目と 4 行目の間がプリアンブルである.

ソースコード 2 .tex ファイルで基本となる構造

```
1    \documentclass[a4paper,12pt]{jsarticle}
2    % Preamble
3
4    \begin{document}
5    % Main text
6
7    \end{document}
```

これから自分が作成する文書において, どのようなパッケージが必要になるかを予め把握しようとするのは得策ではない. 文書作成を進める中で「こういう数学表現を使いたい, L^AT_EX ではどのように書くのだろうか?」というニーズが発生した際に, その都度調べる方が良い. アメリカ数学会 (American Mathematical Society; AMS) がサポートしている `amsmath` パッケージは物理の分野でも重宝されている. 同じ数学表現であっても, 異なるパッケージで異なる記述方法が与えられていることも多々ある. 各自で自分好みの

使いやすいものを見つけていってもらいたい.*2 *3

3 環境

L^AT_EX における環境とは、`\begin{*}` と `\end{*}` という対で囲まれた部分のことを指す。どういう環境かを呼称するために、`{*}` の部分を使って、「* 環境」と言うことがある。例えば、多くの.tex ファイルでは、プリアンプルの直後は `document` 環境になっている。ソースコード 2 で言えば、4 行目と 7 行目の間に文章や数式を入力していくことになる。数式を書く場合には、数式環境を用意する必要がある。代表的な数式環境として、`equation` 環境、`eqnarray` 環境、`align` 環境などがある。例えば、`align` 環境を使う場合は下のようになる。

ソースコード 3 `align` 環境による記述

```
1 \begin{align}
2 % Equation
3 \end{align}
```

4 文書入力ルール

文書の表題には、表題、著者名、著者の所属、作成日時を書く。これらはプリアンプルで、`\title{ 表題 }`、`\author{ 著者名 \\ 著者の所属 }`、`\date{\today}` を記述し、`\begin{document}` のすぐ下で `\maketitle` を書くことで出力される.*4 下のソースコードを参照せよ。

ソースコード 4 表題の作成

```
1 \documentclass[a4paper,12pt]{jsarticle}
2 % Preamble
3 \title{Title}
4 \author{Name\\Institution}
```

*2 時たま、パッケージの組み合わせに起因してコンパイルが出来ないことがある（パッケージ A とパッケージ B を同時に指定するとエラーになる、パッケージ B を指定する場合にはパッケージ A よりも先に指定しなければならない、など）。そういうこともあるのだと頭の片隅に入れておくとどこかで役に立つかもしれない。

*3 例えば、以下のサイトでは数学記号や特殊文字を手書き入力すると、L^AT_EX でそれを再現するために必要なパッケージと使い方が検索される。<https://detexify.kirelabs.org/classify.html>

*4 論文の場合には要旨も書く。

```

5    \date{\today}
6
7    \begin{document}
8    \maketitle
9    % Main text
10   \end{document}

```

通常の文章は `\maketitle` の下に記述していく。L^AT_EX では、%で始まる行はコメントと見なされ、コンパイルしても出力されない。これをコメントアウトと呼ぶ。行の途中に %を書いた場合、そこからその行の最後までがコメントアウトされる。 .tex ファイル（以下ではソースファイルとも呼ぶ）に入力した文章は、それが最も美しく清書されるように L^AT_EX が自動的に改行を入れてくれる。ソースファイル上で改行を行っても出力される清書には改行が反映されない。指定した場所に改行を施したい場合は、ソースファイル上で `\\` を入れる必要がある。また、空行を入れると改段落される。^{*5} 章や節のタイトルは、`\chapter{}`、`\section{}`、`\subsection{}` などで指定できる。章や節の番号は L^AT_EX が自動で割り振ってくれる。^{*6} 箇条書きをする場合、`itemize` 環境や `enumerate` 環境を使うと良い。以下は、`itemize` 環境による記述である。

ソースコード 5 `itemize` 環境による記述

```

1    \begin{itemize}
2        \item \texttt{itemize環境を使う}
3        \item \texttt{itemize環境を使う}
4        \item \texttt{itemize環境を使う}
5    \end{itemize}

```

- `itemize` 環境を使う
- `itemize` 環境を使う
- `itemize` 環境を使う

以下は、`enumerate` 環境による記述である。こちらの環境では連番が付される。

ソースコード 6 `enumerate` 環境による記述

```

1    \begin{enumerate}
2        \item \texttt{enumerate環境を使う}

```

^{*5} 改ページしたい場合は、改ページしたい場所で `\clearpage` と書けばよい。

^{*6} これらの番号を自分で変更したい場合には、`\setcounter{}` を使う。

```

3      \item \texttt{enumerate環境を使う}
4      \item \texttt{enumerate環境を使う}
5      \end{enumerate}

```

1. `enumerate` 環境を使う
2. `enumerate` 環境を使う
3. `enumerate` 環境を使う

いずれの環境でも、`\item` によって新しい箇条書きの項目を開始する。

5 表の作成

表の作成には、`table` 環境と `tabular` 環境を使う。具体例として、表 1 とそのソースコード 7 を見よ。`\hline` によって横線が一本引かれる。`\hline \hline` と書けば二重の横線を引くことができる。表の各行において、それぞれの要素の間は`&`で区切る。行の最後には`\\`を書く。表 1 では、各行が四つの要素から構成されている。そして、各要素は中央揃えで表示されている。これは、ソースファイル上の`\begin{tabular}{c|c|c|c}`の部分で制御されている。`{c|c|c|c}`の`|`によって縦線が出力される。`|`を消すと縦線が消えた表になる。また、`c`を`l`, `r`に変更すると、左揃え、右揃えに変えることができる。表に対するキャプションは`\caption{}`で記述する。

ソースコード 7 表 1

```

1      \begin{table}[H]
2          \centering
3          \caption{Gauge boson}
4          \begin{tabular}{c|c|c|c}
5              \hline
6              Name & Mass (GeV) & Electronic charge & Interaction \\
7              \hline\hline
8              Photon & $0$ & $0$ & Electromagnetic \\
9              Gluon & $0$ & $0$ & Strong \\
10             $W$ boson & $89.3$ & $\pm 1$ & Electromagnetic, Weak,
11                 Gravity \\
12             $Z$ boson & $93$ & $0$ & Weak, Gravity \\
13             Graviton & $0$ & $0$ & Gravity \\
14             \hline

```

```

14         \end{tabular}
15         \label{tab:sample}
16     \end{table}

```

表 1 Gauge boson

Name	Mass (GeV)	Electronic charge	Interaction
Photon	0	0	Electromagnetic
Gluon	0	0	Strong
W boson	89.3	± 1	Electromagnetic, Weak, Gravity
Z boson	93	0	Weak, Gravity
Graviton	0	0	Gravity

6 図の挿入

まずは gnuplot や Python で作成した図 (.pdf や .eps) を fig ディレクトリに格納する。例として, fig ディレクトリに格納した sample.pdf を文章中に挿入する。ソースファイル上で fig 環境を作り, その中で `\includegraphics{fig/sample.pdf}` と書けば図を挿入できる (図 1)。表の場合と同様, 図の説明を `\caption{}` で記述できる。実際のソースコードでは, `\includegraphics[scale=1]{fig/sample.pdf}` となっているが, `[scale=1]` は図の大きさを調整するためのオプションである。L^AT_EX では, 図を表示する場所が自動的に調整されるが, `\begin{figure}` にオプションを付すことで, その場所を自分で操作することもできる。例えば, `\begin{figure}[h]` とすると, ソースファイル上で `\includegraphics` したその場所に図を優先的に表示させることができる。^{*7}

ソースコード 8 図 1

```

1     \begin{figure}
2         \centering
3         \includegraphics[scale=1]{fig/sample.pdf}
4         \caption{Trigonometric functions}
5         \label{fig:sample}

```

^{*7} その場所に強制的に表示させたい場合は, `here.sty` を使い, `\usepackage{here}` をプリアンブルに書いた上で, `\begin{figure}[H]` とする。

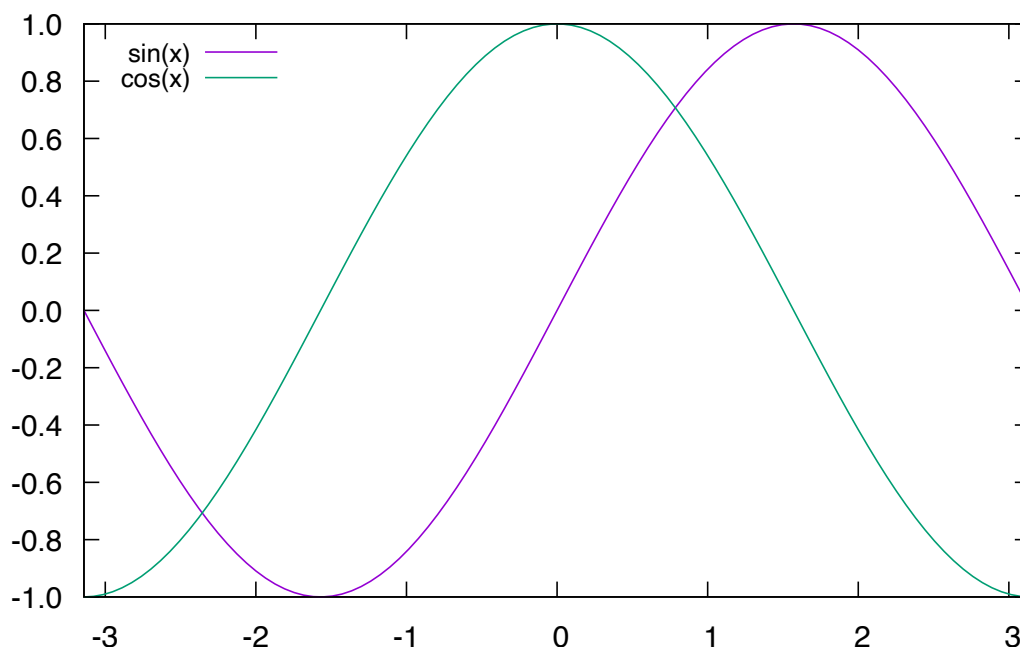


図1 Trigonometric functions

7 数式，図，表の参照

LaTeX では、数式、図、表には番号が自動的に割り振られ、`\label{*}` と `\ref{*}` のペアでこれらの番号を容易に参照することができる。ここで、`\label{*}` と `\ref{*}` のペアにおいて、`{*}` には自分の好きな名前を与えることができるが、数式に対しては `{eq:*}`、表に対しては `{tab:*}`、図に対しては `{fig:*}`、のように参照名をつけるのが慣習である。例えば、数式を式番号で参照する場合、参照したい数式が記述されている環境内に `\label{eq:*}` を書いておく。そして、文章中で `\ref{eq:*}` と書けばその式番号が表示される。実際には、式番号には丸括弧 () を付けることが多いため、`(\ref{eq:*})` とするか、数式専用の参照方法として `\eqref{eq:*}` を用いることが多い。実際に以下の数式を参照してみると次のようになる。

ソースコード 9 式 (7.1)

1 `\begin{align}`

```

2      \label{eq:sample}
3      f(x)=x
4      \end{align}

```

$$f(x) = x \tag{7.1}$$

上の式は式 (7.1) として参照される。この式よりも前に別の式を書き足しても、 \LaTeX が勝手に新しい式番号をつけ、 $\text{\label{*}}$ と $\text{\ref{*}}$ によって新しい式番号が参照される。式番号そのものをソースファイル上で打つ（ハードコーディング）のは避ける。ハードコーディングは間違いの元である。また、異なる式には異なる参照名を与えること。

8 いくつかの数学表現

8.1 諸注意と数学関数

数式を書くためには数式環境を使う。例えば、 align 環境を使う場合、 $\text{\begin{align}}$ と $\text{\end{align}}$ の間に数式を入力していく。もし、文章中で数式を書く場合には $\text{\$}$ で数式を挟む必要がある。^{*8}

数式環境では、文字はすべて斜体で表示される。例えば、ソースファイル上で $\text{\$sin x\$}$ と書くと、 $\sin x$ となる。このような数学関数は斜体ではなく、立体的に表示するのがルールである。多くの数学関数は、 \ に数学関数の名前を続けると立体的に表示される。今の場合、 $\text{\$sin x\$}$ とすれば、 $\sin x$ と出力される。注意すべきは、 \sin x のように \sin と x の間にはスペースが必要なことである。スペースなしで \sinx と書いてしまうと、 \sinx というコマンドとして見なされる。指定しているパッケージ内に存在しないコマンドを入力してしまうと当然エラーになる。主な数学関数を表 2 に示す。

表 2 主な数学関数

コマンド	出力	コマンド	出力	コマンド	出力
\cos	cos	\sin	sin	\tan	tan
\cosh	cosh	\sinh	sinh	\tanh	tanh
\exp	exp	\ln	ln	\log	log
\sec	sec	\csc	csc	\cot	cot
\lim	lim	\sup	sup	\inf	inf

^{*8} インライン数式モードなどと呼ばれる。

なお、物理の分野では、物理的に意味のある量は斜体で記述するが、物理的な意味を持たない定数などは立体で書くというルールがある。例えば、素電荷はよく e で表すのに対し、Napier 数は e と表記する。数式環境でアルファベットを立体にするには、`\mathrm{アルファベット}` とすれば良い。他にも、虚数単位は i のように立体で書く。微分記号の d も立体で書く。例えば、指数関数 e^x の微分は、

ソースコード 10 式 (8.1)

```

1 \begin{align}
2 \quad \frac{\mathrm{d}}{\mathrm{d}x}\mathrm{e}^x
3 \quad = \mathrm{e}^x
4 \end{align}

```

$$\frac{d}{dx}e^x = e^x \quad (8.1)$$

のように書く。なお、ソースコード 10 では見やすさを優先してインデントや改行が入っているが、これらがなくても式 (8.1) と同じ出力が得られる。実際、以下のように書いても全く問題ない。

ソースコード 11 式 (8.1)

```

1 \begin{align}
2 \quad \frac{\mathrm{d}}{\mathrm{d}x}\mathrm{e}^x=\mathrm{e}^x
3 \end{align}

```

以下のソースコードでは、常に見やすさを優先してインデントや改行が適宜入れている。

8.2 ギリシャ文字

ギリシャ文字は大変よく使う。表 3 がギリシャ文字の一覧である。大文字にするには、コマンドの頭文字だけ大文字にする。例えば、 Ψ を表示するためには `\Psi` と書く。ただし、全てのギリシャ文字がこのルールに従っているわけではなく、通常のアスタリスクの大文字で代用する場合もある。例えば、 α の大文字は A であり、これは単に A と打つだけで良い。また、いくつかのギリシャ文字については、綴りの前に `var` を付けることで書体の変更が可能である。例えば、単に `\epsilon` と書くと ϵ となるが、`\varepsilon` と書くと ε になる。

表3 ギリシャ文字

コマンド	出力	コマンド	出力	コマンド	出力
<code>\alpha</code>	α	<code>\iota</code>	ι	<code>\rho</code>	ρ
<code>\beta</code>	β	<code>\kappa</code>	κ	<code>\sigma</code>	σ
<code>\gamma</code>	γ	<code>\lambda</code>	λ	<code>\tau</code>	τ
<code>\delta</code>	δ	<code>\mu</code>	μ	<code>\upsilon</code>	υ
<code>\epsilon</code>	ϵ	<code>\nu</code>	ν	<code>\phi</code>	ϕ
<code>\zeta</code>	ζ	<code>\xi</code>	ξ	<code>\chi</code>	χ
<code>\eta</code>	η	<code>o</code>	o	<code>\psi</code>	ψ
<code>\theta</code>	θ	<code>\pi</code>	π	<code>\omega</code>	ω

8.3 数学記号，特殊文字

数式環境の中で使うことのできる数学記号，特殊文字のいくつかを表4に示す．

表4 代表的な数学記号，特殊文字

コマンド	出力	コマンド	出力
<code>\equiv</code>	\equiv	<code>\approx</code>	\approx
<code>\simeq</code>	\simeq	<code>\sim</code>	\sim
<code>\leq</code>	\leq	<code>\geq</code>	\geq
<code>\lesssim</code>	\lesssim	<code>\gtrsim</code>	\gtrsim
<code>\rightarrow</code>	\rightarrow	<code>\mapsto</code>	\mapsto
<code>\rightarrowtail</code>	\rightarrowtail	<code>\Rrightarrow</code>	\Rrightarrow
<code>\leftarrowtail</code>	\leftarrowtail	<code>\Lleftarrow</code>	\Lleftarrow
<code>\leftrightarrow</code>	\leftrightarrow	<code>\Leftrightarrow</code>	\Leftrightarrow
<code>\pm</code>	\pm	<code>\mp</code>	\mp
<code>\propto</code>	\propto	<code>\infty</code>	∞
<code>\times</code>	\times	<code>\otimes</code>	\otimes
<code>\partial</code>	∂	<code>\nabla</code>	∇
<code>\dagger</code>	\dagger	<code>\hbar</code>	\hbar
<code>\cdots</code>	\cdots	<code>\cdots</code>	\cdots

8.4 添え字

上付きの添え字を書くためには \wedge を使う。下付きの添え字には $_$ を使う。

ソースコード 12 式 (8.2)

```
1 \begin{align}
2 2^{2024}
3 \end{align}
```

$$2^{2024} \quad (8.2)$$

ソースコード 13 式 (8.3)

```
1 \begin{align}
2 a_{1}, a_{2}, a_{3}
3 \end{align}
```

$$a_1, a_2, a_3 \quad (8.3)$$

上付き、下付きの添え字を組み合わせる場合はどちらを先に書いても構わない。

ソースコード 14 式 (8.4)

```
1 \begin{align}
2 g^{\{\mu\}}_{\{\nu\}}, g_{\{\nu\}}^{\{\mu\}}
3 \end{align}
```

$$g_{\nu}^{\mu}, g_{\nu}^{\mu} \quad (8.4)$$

なお、ソースコード上で上付き、下付きの添え字は必ずしも $\{\}$ で括る必要はないが、次の例から注意が必要なことが分かるだろう。

ソースコード 15 式 (8.5)

```
1 \begin{align}
2 g^{\mu\nu}, g^{\mu}\nu
3 \end{align}
```

$$g^{\mu\nu}, g^{\mu}\nu \quad (8.5)$$

8.5 分数

分数は `\frac{ 分子 }{ 分母 }` と書く.

ソースコード 16 式 (8.6)

```
1 \begin{align}
2 \tan x = \frac{\sin x}{\cos x}
3 \end{align}
```

$$\tan x = \frac{\sin x}{\cos x} \quad (8.6)$$

8.6 大型演算子

和は `\sum`, 積は `\prod`, 積分は `\int` で書く. これらの上端, 下端は上付き, 下付き添え字で書く.

ソースコード 17 式 (8.7)

```
1 \begin{align}
2 \sum_{k=1}^{\infty} \frac{1}{k^2}
3 = \frac{\pi^2}{6}
4 \end{align}
```

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6} \quad (8.7)$$

ソースコード 18 式 (8.8)

```
1 \begin{align}
2 \prod_{k=1}^{\infty} \frac{(2k)^2}{(2k-1)(2k+1)}
3 = \frac{\pi^2}{2}
4 \end{align}
```

$$\prod_{k=1}^{\infty} \frac{(2k)^2}{(2k-1)(2k+1)} = \frac{\pi^2}{2} \quad (8.8)$$

ソースコード 19 式 (8.9)

```
1 \begin{align}
2 \int^{2\pi}_{0}\mathrm{d}x\cos x
3 = 0
4 \end{align}
```

$$\int_0^{2\pi} dx \cos x = 0 \quad (8.9)$$

8.7 根号

根号は `\sqrt{}` で書く. n 乗根の場合は `\sqrt[n]{}` と書く.

ソースコード 20 式 (8.10)

```
1 \begin{align}
2 \sqrt{1+x}
3 \end{align}
```

$$\sqrt{1+x} \quad (8.10)$$

ソースコード 21 式 (8.11)

```
1 \begin{align}
2 \sqrt[4]{1+x}
3 \end{align}
```

$$\sqrt[4]{1+x} \quad (8.11)$$

8.8 ベクトル

ベクトルは太文字を使う. 太文字にするには, `bm` パッケージを指定した上で `\bm{}` を使う.

ソースコード 22 式 (8.12)

```
1 \begin{align}
2 \bm{x}
3 \end{align}
```

$$\boldsymbol{x} \quad (8.12)$$

もし矢印でベクトルを表したい場合は `\vec{}` を使う.

ソースコード 23 式 (8.13)

```

1 \begin{align}
2   \vec{x}
3 \end{align}

```

$$\vec{x} \quad (8.13)$$

ベクトルの成分をあらわに書く場合は, `pmatrix` 環境を使うと良い. 数式環境の中に `pmatrix` 環境を作ることに注意.

ソースコード 24 式 (8.14)

```

1 \begin{align}
2   \bm{x} =
3   \begin{pmatrix}
4     x_1 \\
5     x_2 \\
6     x_3
7   \end{pmatrix}
8 \end{align}

```

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (8.14)$$

また, `bmatrix` 環境を使うと,

ソースコード 25 式 (8.15)

```

1 \begin{align}
2   \bm{x} =
3   \begin{bmatrix}
4     x_1 \\
5     x_2 \\
6     x_3
7   \end{bmatrix}
8 \end{align}

```

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (8.15)$$

となる.

8.9 文字装飾

アルファベットに対する装飾をいくつか紹介する. ハット記号は `\hat{}`, チルダ記号は `\tilde{}`, ドット記号は `\dot{}` を使う.

ソースコード 26 式 (8.16)

```
1 \begin{align}
2   \hat{H}, \tilde{I}, \dot{J}
3 \end{align}
```

$$\hat{H}, \tilde{I}, \dot{J} \quad (8.16)$$

なお, ソースコード 2 行目で使われている `~` はスペースを表す.

物理の分野では, ドット記号で時間微分を表す.

ソースコード 27 式 (8.17)

```
1 \begin{align}
2   \dot{x} = \frac{\mathrm{d}}{\mathrm{d}t}x
3 \end{align}
```

$$\dot{x} = \frac{\mathrm{d}}{\mathrm{d}t}x \quad (8.17)$$

8.10 行列

すでにベクトルのところで使ったが, 行列を書く場合にも `pmatrix` 環境を使うと良い. 数式環境の中に `pmatrix` 環境を作ることには注意. 表の作成方法と類似している.

ソースコード 28 式 (8.18)

```
1 \begin{align}
2   A =
3   \begin{pmatrix}
```

```

4          a_{11} & a_{12} \\
5          a_{21} & a_{22}
6      \end{pmatrix}
7  \end{align}

```

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad (8.18)$$

また, `bmatrix` 環境を使うと,

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (8.19)$$

となる. 行列式の場合は `vmatrix` 環境を使うと良い.

$$A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \quad (8.20)$$

`Vmatrix` 環境にすると,

$$A = \left\| \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \right\| \quad (8.21)$$

となる.

8.11 括弧の大きさ

以下の数式を見てみよう.

ソースコード 29 式 (8.22)

```

1  \begin{align}
2  \int_0^n \mathrm{d}x (1 - \frac{x}{n})^n x^{z-1}
3  \end{align}

```

$$\int_0^n dx \left(1 - \frac{x}{n}\right)^n x^{z-1} \quad (8.22)$$

ここで, ソースコード 4 行目の丸括弧を表す `()` のペアを `\left(\right)` に置き換えると, 丸括弧の中身に応じて丸括弧の大きさを自動的に調整してくれる.

ソースコード 30 式 (8.23)

```

1  \begin{align}
2  \int_0^n \mathrm{d}x \left(1 - \frac{x}{n}\right)^n x^{z-1}
3  \end{align}

```

$$\int_0^n dx \left(1 - \frac{x}{n}\right)^n x^{z-1} \quad (8.23)$$

丸括弧に限らず, $\{ \}$ や $[]$ についても `\left` と `\right` を付すことでその大きさが自動的に調整される.

ソースコード 31 式 (8.24)

```

1 \begin{align}
2 \quad \left[
3 \quad \quad \left\{
4 \quad \quad \quad \left(
5 \quad \quad \quad \quad \frac{1}{x}
6 \quad \quad \quad \right)
7 \quad \quad \right\}
8 \quad \right]
9 \end{align}
```

$$\left[\left\{ \left(\frac{1}{x} \right) \right\} \right] \quad (8.24)$$

8.12 Dirac のブラケット記法

L^AT_EX で Dirac のブラケット記法を使う方法はいくつか考えられる. ここでは, `braket` パッケージを使う方法を紹介する. `braket` パッケージでは, `\braket{}`, `\bra{}`, `\ket{}` で簡単にブラケット記法を使うことができる.

ソースコード 32 式 (8.25)

```

1 \begin{align}
2 \quad \braket{\psi|\hat{A}|\psi}
3 \end{align}
```

$$\langle \psi | \hat{A} | \psi \rangle \quad (8.25)$$

ソースコード 33 式 (8.26)

```

1 \begin{align}
2 \quad \int \mathrm{d}x \ket{x} \bra{x}
3 \end{align}
```

$$\int dx |x\rangle \langle x| \quad (8.26)$$

8.13 数式環境内での改行

式変形を記述する場合，数式環境内で改行を行うことがある．改行は `\\` で行えば良いが，単に `\\` だけ使って下のように書くとあまり綺麗な出力にならない．

ソースコード 34 `\\` だけを使った場合

```

1 \begin{align}
2 \int^1_0 \mathrm{d}x \log(1+x) = [(1+x)\log(1+x)]^1_0
3 - \int^1_0 \mathrm{d}x (1+x) \frac{1}{1+x} \\
4 = 2\log 2 - 1
5 \end{align}
```

$$\int_0^1 dx \log(1+x) = [(1+x) \log(1+x)]_0^1 - \int_0^1 dx (1+x) \frac{1}{1+x} \quad (8.27)$$

$$= 2 \log 2 - 1 \quad (8.28)$$

式番号を二つも付す必要はないし，`=` の場所も揃えた方がより見栄えがするだろう．式番号を消すには `\nonumber` を使う．`=` の場所を揃えるには，ソースコード上で二つの`=`の前に`&`を付す．

ソースコード 35 式 (8.29)

```

1 \begin{align}
2 \int^1_0 \mathrm{d}x \log(1+x)
3 &= [(1+x)\log(1+x)]^1_0 - \int^1_0 \mathrm{d}x (1+x)
4 \frac{1}{1+x} \nonumber \\
5 &= 2\log 2 - 1
6 \end{align}
```

$$\int_0^1 dx \log(1+x) = [(1+x) \log(1+x)]_0^1 - \int_0^1 dx (1+x) \frac{1}{1+x} = 2 \log 2 - 1 \quad (8.29)$$

改行を跨いで`&`が付されている場合，`&`が付された場所を自動的に揃えてくれる．次のような揃え方も可能である．

```

1 \begin{align}
2 & \int_0^1 \mathrm{d}x \log(1+x) \quad \text{\nonumber \\}
3 & = [(1+x)\log(1+x)]_0^1 - \int_0^1 \mathrm{d}x (1+x) \frac{1}{1+x} \quad \text{\nonumber \\}
4 & = 2\log 2 - 1
5 \end{align}

```

$$\begin{aligned}
 & \int_0^1 dx \log(1+x) \\
 & = [(1+x)\log(1+x)]_0^1 - \int_0^1 dx (1+x) \frac{1}{1+x} \\
 & = 2\log 2 - 1
 \end{aligned} \tag{8.30}$$

9 参考文献

参考文献は文末に記載する。参考文献の番号も \LaTeX が自動的に割り振ってくれる。 \LaTeX で参考文献を記載する方法にはいくつかある。例えば、`thebibliography` 環境を使う方法がある。ただし、この方法では文献情報をすべて手打ち入力しなければならず、参考文献の数が増えるほど億劫な作業を強いられる。そこで、本稿では BibTeX という文献管理データベースを使って、参考文献の情報を管理する方法を採用する。

下のソースコードは、とある教科書の情報を BibTeX で記述したものである。どのような情報が含まれているかは一目瞭然だろう。このような BibTeX による文献情報は、自分で手打ちしなくても、Google Scholar や大学図書館の Tulips Search などを使えば簡単に取得できる。^{*9} 例えば、Google Scholar で文献を検索し、「引用」のボタンを押せば、BibTeX を選択することができる。そうすれば、下のようなソースコードをコピーできる。コピーした情報は、bib ディレクトリの中に bib ファイルを作り、そこに貼り付けばよい。ここでは、bib ディレクトリの中に sample.bib というファイルを作って、そこに下のソースコードを貼り付けたとしよう。

^{*9} 学術論文の場合、出版社のサイトからも BibTeX による論文情報を取得できることがほとんどである。探せば必ず手に入るという前提でいた方がよい。

ソースコード 37 BibTeX による文献情報の記述

```
1 @book{peskin2018introduction,  
2     title={An introduction to quantum field theory},  
3     author={Peskin, Michael E},  
4     year={2018},  
5     publisher={CRC press}  
6 }
```

後は, `document` 環境の一番下 (つまり `\end{document}` のすぐ上) で次のように書く.

ソースコード 38 .bib ファイルの読み込み

```
1 \bibliographystyle{unsrt}  
2 \bibliography{bib/sample}
```

1 行目で参考文献の表示スタイルを指定しており, `sample.bib` の読み込みは 2 行目で行われている. ここでは, `unsrt` というスタイルを例として用いた. 以上で準備完了である. ソースコード 37 の文献を引用するには, 文章中で `\cite{peskin2018introduction}` と書くだけでいい. そうすれば, 参考文献 [1] のようになる. 日本語の文献も参考文献 [2] のように引用できる. なお, 例えば Tulips Search で文献情報を取得する場合, ソースコード 37 の 1 行目の `peskin2018introduction`, に該当する部分が空欄になっていることがある. この部分は `\cite{}` で引用する時に使うマーカーであり, 自由に設定してよい.^{*10} 好きなマーカーを設定した後, 最後にカンマ (,) を入れるのを忘れないように.

最後の注として, 複数の `.bib` ファイルを読み込ませることもできる. 例えば, `bib` ディレクトリに `textbook.bib` と `paper.bib` という `.bib` ファイルを作ったとしよう. これらを読み込ませる場合には, 以下のようにするだけでいい.

ソースコード 39 複数の.bib ファイルの読み込み

```
1 \bibliographystyle{unsrt}  
2 \bibliography{bib/textbook,bib/paper}
```

参考文献

- [1] Michael E Peskin. *An introduction to quantum field theory*. CRC press, 2018.
- [2] 砂川重信. **理論電磁気学 第 3 版**. 紀伊國屋書店, 1999.

^{*10} すでにマーカーが設定されている場合はあえて変更しなくてよい.