

L^AT_EX 文書の書き方

Copyright©KAKEHI Katsuhiko 1996-1998

All Rights Reserved

1 全体の構成

L^AT_EX 文書は、つぎの形に書く。

```
\documentstyle[<書式追加指定>]{jarticle}
\title{<題目>}
\author{<著者名>}
\date{<日付>}
<定義>
\begin{document}
<本体>
\end{document}
```

最初の行は、文書全体の形態を指定する。文書の形態として用意されているものには、つぎのものがある。

`article` 大部でない報告、論文など。章の単位をもたない。日本語文書の場合は、`jarticle` と指定する。

`report` 大部の報告、論文など。章の単位をもつ。表紙、概要、参考文献などがそれぞれ独立したページになる。日本語文書の場合は、`jreport` と指定する。

`book` 本。

通常提出するレポートなどは、`jarticle` で十分である。卒業論文の場合は、`jreport` の形態をとるのがよい。この資料では、`jarticle` を対象として話を進める。

1.1 書式追加指定

書式追加指定には、標準の設定に代えて特別な形態を指定したいときに、つぎのような指定を“,”で区切って並べる。

`11pt`, `12pt` 標準字体の寸法を指定する。標準は `10pt`。

`twoside` 両面印刷用に奇数ページ、偶数ページで配置を変えることを指定する。標準は `oneside`。

`twocolumn` 2段組を指定する。標準は `onecolumn`。

`titlepage` 題目、著者などを示す表紙を単独のページとし、概要があればこれも単独のページとする。標準は `notitlepage`。

1.2 題目など

文書の題目、著者、(作成)日付は、それぞれ<題目>、<著者>、<日付>のところに書き込む。それぞれについて、(最終印字の際に)複数の行に分けて表示したい場合は、行の切れ目を `\\` を書いて指定する。

例 `\title{変数変換について \\ - その1 - }`

日付として L^AT_EX で処理した日を表示したいときは、`\today` と書けばよい。

1.3 定義

文書全体にわたっての各種の寸法（上下の余白、左の余白など）は、標準値が設定されているが、これを変更したいときにその指定をここに書き並べる。（用紙の左端から1インチ、上端から1インチは余白として先に取られている。）

<code>\topmargin</code>	最上部の余白	<code>\oddsidemargin</code>	左端の余白（奇数ページ）
<code>\headheight</code>	ヘッディングの高さ	<code>\evensidemargin</code>	左端の余白（偶数ページ）
<code>\headsep</code>	ヘッディングと本体との間隔	<code>\textwidth</code>	本体の幅
<code>\textheight</code>	本体の高さ		
<code>\footskip</code>	フッティング部の全体の高さ		
<code>\footheight</code>	フッティングそのもの高さ		

例 `\textwidth 170mm`

自分で指令や環境を定義して使う場合には、この部分にその定義を書き並べる。

2 本体

本体は、つぎの形に書く。

```
[ \maketitle ]
[ < 概要 > ]
< 本文 >
[ < 参考文献 > ]
```

表紙情報（題目・著者名・日付）を書き出す指定 `\maketitle`、概要、参考文献は、必要ならそれぞれこの位置に書く。

2.1 概要

概要は、つぎの形に書く。

```
\begin{abstract}
< 文章 >
\end{abstract}
```

概要は、その文書の内容を簡潔に紹介する。通常は、200字から400字程度で書く。

2.2 参考文献

その文書で直接間接に参照したり引用したりした文献の一覧を、つぎの形に書く。

```
\begin{thebibliography}{< 幅指定 >}
{ \bibitem{< 参照略号 >} < 文献 > }
\end{thebibliography}
```

`\bibitem` として書き並べたものに通し番号がつく。幅指定には、この通し番号を表示するのに必要な桁数だけ、9を書き並べる。（文献数1桁台なら1個、2桁台なら2個）

参照略号は、本文の中でその文献を参照するときに使う略号を書く。略号は、英数字でつづる。参照する箇所では、`\cite{< 参照略号 >}` と書けばよい。

文献の書き方には、いろいろな流儀がある。要は、その文献を見てみたいと思ったときに、確実に入手できるだけの情報を与えることに尽きる。

本 本をまるごと参考とする場合は、少なくとも、著者名、書名、出版社、出版年を書く。

記事 雑誌や本の中の記事を参考とする場合は、少なくとも、著者名、記事名、それが載っていた書籍名（雑誌の場合は、巻・号も書く）、その記事の掲載ページ、出版年を書く。

2.3 本文

本文は、一般に、つぎの形に節分けして書く。

```
{ \section{<見出し>}
  <文章>
  { \subsubsection{<見出し>}
    <文章>
    { \subsubsection{<見出し>}
      <文章>
    }
  }
}
```

`\section` と指定したものには、1、2、... と順に節番号がつく。`\subsection` と指定したものには 1.1 などと節番号がつき、`\subsubsection` と指定したものには 1.1.1 などと節番号がつく。

特定の節の節番号を文章中で参照する必要があるときは、節の見出しを書いた行の末尾に`\label{<参照略号>}`を書き、引用する場所に`\ref{<参照略号>}`と書けばよい(2.2 参照)。

3 文章

それでは、いよいよ文章を書き並べるときの規則を見て行くことにしよう。

3.1 記号

これまでに見てきたように、 \LaTeX では、そのまま印字される文面とともに文書の構造や印字形態などの指示が混在している。このような指示を表わすための記号には、つぎのものがある。

<code>\</code>	指示の開始を示す
<code>{, }</code>	指示の引数などのまとまりを括る
<code>%</code>	注釈の開始。そこから行末までを無視する
<code>~</code>	そこで改行が起きないように指定する
<code>&</code>	表形式などでの項目の区切りを示す
<code>\$</code>	数式相との切り換えを示す
<code>#</code>	指示を新たに定義する際の引数を示す

なお、`[と]` も指示の補助引数を括るために使うが、そうした使い方以外では、その記号そのものが印字される。つぎの記号は、数式相の中でだけ使うことができる。

`_ ^ | < >`

こうした特殊な文字のうち、つぎのものは、前に `\` を付けることで、その文字そのものを表わすことができる。

`{ } % & $ # _`

なお、`' と '` は、つぎのように特別な扱いを受ける。

入力	印字	入力	印字
<code>'</code>	<code>'</code>	<code>‘</code>	<code>“</code>
<code>,</code>	<code>,</code>	<code>’</code>	<code>”</code>

さらに、`'` は、数式相(4 参照)の中では`'`(微分記号)として扱われるので注意すること。

3.2 段落分け

文章は、段落分けして書く。段落の切れ目は、空の行で示す。(空の行を何行続けても、一つの段落の切れ目を示すにすぎない。)

逆に、改行は、単語の切れ目を示す働きだけをもつ。日本語の文章では、“単語” 分けをすることはないから改行は(二つ続ければ空の行が生じて段落分けが生じるが)、特段の働きをもたない。

多くの場合に、印字の際に強制的に改行をしたいなら、`\\` と書けばよい。

L^AT_EX は、英文 (欧文) を基本として作られているから、英文での文を終える記号 (., !, ?, :) は、単語を区切る空白よりも大きな空白があく。ただし、人名など省略にも . が使われることを配慮して、英大文字の直後の . は単語の終わりとしてだけ働く。英大文字で文が終わるときには、その . の後ろに文末としての空白を置くように指示する必要がある。こうした場合を含め、特別に空白を置くことを指示するものにつぎのものがある。

```
\@ 文末の空白
\l 単語を区切る空白
\, ‘ や “ の直後、 ’ や ” の直前に置く空白
```

同様に、英文としての記号の使い方に合わせた、つぎの約束がある。

- - を続けると、その個数に見合ったダッシュ記号として印字される。たとえば、--- は—となる。
- \ldots は、... と印字される。

3.3 引用

文章の中で、他の文献の記述から文や段落を引用する場合には、つぎのいずれかの形で書く。それぞれ引用した部分が、字下げされて表示される。

- `\begin{quote} <引用内容> \end{quote}`
比較的短い文章を引用する場合に使う。引用内容の中で強制的に改行したいときは、`\\` を書く。
- `\begin{quotation} <引用内容> \end{quotation}`
引用が複数の段落にわたるような長い文章を引用する場合に使う。

3.4 箇条書き

箇条書きには、三つの形態が用意されている。

3.4.1 順不同の列記

いくつかの項目、あるいは箇条を順不同に列記する場合、その見出しとして、たとえば “・” といった記号を使うものである。この形態での列記は、つぎの形に書く。

```
\begin{itemize}
{ \item <箇条内容> }
\end{itemize}
```

3.4.2 付番列記

いくつかの項目、あるいは箇条を番号を付けて列記する場合は、つぎの形に書く。

```
\begin{enumerate}
{ \item <箇条内容> }
\end{enumerate}
```

その番号を参照したい場合のやり方は、節番号の参照と同じである。

3.4.3 見出し付列記

それぞれの項目 (あるいは箇条) に見出しをつけて列記する場合には、つぎの形に書く。

```
\begin{description}
{ \item[ <見出し語> ] <箇条内容> }
\end{description}
```

3.5 表の形式

3.5.1 タブ

文節の位置や、行頭の位置を上の行との関係で特定の位置に揃えたいことがある。このときには、タブの機能を使う。タブを使うには、つぎの形に書く。

```
\begin{tabbing} < タブ利用文章 > \end{tabbing}
```

タブ利用文章の中では、つぎの指定が使える。

```
\=      タブ位置をその位置に設定する  
\>     次のタブ位置へ進む  
\&     改行して次の行へ進む  
\kill  現在行を捨てる ( 印字しない )
```

\kill は、タブ位置を設定しておくために、位置決めを行うためだけに行を書くのに使う。

```
例 \begin{tabbing}                               ⇒ いくつかの例を示す。  
   書き出し \= 例 \kill                          if 条件 then  
   いくつかの例を示す。\\                       真のとき実行する文  
   \> \verb+if+ \= 条件 \verb+then+ \\          else  
   \> \> 真のとき実行する文 \\                  偽のとき実行する文  
   \> \verb+else+ \\  
   \> \> 偽のとき実行する文  
\end{tabbing}
```

3.5.2 表形式

表の形に整えたい場合は、つぎの形に書く。

```
\begin{tabular}{ < 列指定 > }  
  { { \hline }  
  < 列項目 > { & < 列項目 > } \\  
  }  
  { \hline }  
\end{tabular}
```

ここで、\hline は、横罫線を引くことを指示する。表の 1 行分の終わりは、\\ で示す。ただし、表の最後の行の下に横罫線を引かないときは、最後の行に \\ をつけない。

列指定には、それぞれの列の列項目を左詰め (l)、中央配置 (c)、右詰め (r) のどれにするかを順に書く。同時に縦罫線を入れる位置があれば | で指定する。

当然のことながら、実際に各行の内容を指定する際の & の個数は、列の個数から 1 減じたものに等しくなければならない。

```
例 \begin{tabular}[t]{l|c|r}\hline ⇒  
   アメリカ合衆国 & \$ & 1.00 \\  
   日本 & Yen & 113.00 \\  
   ドイツ & DM & 1.63  
\end{tabular}
```

アメリカ合衆国	\$	1.00
日本	Yen	113.00
ドイツ	DM	1.63

この例で、[t] と指定してあるのは、他との位置合せを最上行に合わせることを意味する。

3.6 L^AT_EX 文書そのものの表示

L^AT_EX 文書そのものの一部をそのままの形で表示したいときには、つぎの書き方がある。

- \verba < 行の一部 > α
 ここで、α は、空白、英字以外の文字。
- \verb*α < 行の一部 > α
 ここで、α は、空白、英字以外の文字。 < 行の一部 > の中の空白が `␣` として印字される。
- \begin{verbatim} < 文書の一部 > \end{verbatim}
 < 文書の一部 > が複数行にまたがっていても、その改行も含め、そのままの形で印字される。

- `\begin{verbatim}` < 文書の一部 > `\end{verbatim}`
 < 文書の一部 > が複数行にまたがっていても、その改行も含め、そのままの形で印字される。< 文書の一部 > 中の空白は `\` と印字される。

これらの指示は、絶対的なものであり、`\verb α` や `\verb* α` で始めると次に文字 α が来るまで、`\begin{verbatim}` や `\begin{verbatim}` で始めると次に `\end{verbatim}` や `\end{verbatim}` が来るまで、たとえその中に `\` で始まる (ふつつなら) 指示 (と解釈されるもの) があっても解釈されることなく、そのままの形で印字される。

4 数式

数式としての扱いを受ける部分を数式相という。数式相は、つぎの形で指定する。

- `$ < 数式 > $`
- `\(< 数式 > \)`
 これら二つは、文章の行の中にその数式を入れ込むことを指示する。数式は、略式で印字される。
- `$$ < 数式 > $$`
- `\[< 数式 > \]`
 これら二つは、文章から行を分けて数式を印字することを指示する。

例 同じ `$n/2$` でも、⇒ 同じ `n/2` でも、`$$n/2$$` を使うと
`\verb+$$n/2$$+` を使うと
`$$n/2$$` と 1 行とって表示される。`n/2`

と 1 行とって表示される。

数式相は、その数式が 1 行に収まることを前提としている。等式を次々に変形してしていくなど、複数行にわたって数式を書き並べるための特別な書き方が用意されている。これについては、あとで触れる。

4.1 数式相での原則

数式相の中では、空白は何の意味ももたない。英字は、それぞれ 1 字ずつ独立の記号として扱われる。たとえば、`fg` と書いても、`$f g$` と書いても、`fg` と印字される。

数式の印字は、すべて \LaTeX が適宜行ってくれるが、空白の空け方を強制的に指示したいことも起きる。この目的に、つぎのものが用意してある。

`\,` 極小の空白 `\:` 中間幅の空白
`\!` 空白を減らす `\;` 大きめの空白

4.2 記号

数式の中で使うさまざまな記号が用意されている。

4.2.1 ギリシア文字

ギリシア文字にはつぎのものが用意されている。

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>delta</code>	ϵ	<code>\epsilon</code>
ζ	<code>\zeta</code>	η	<code>\eta</code>	θ	<code>\theta</code>	ι	<code>\iota</code>	κ	<code>\kappa</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>	ξ	<code>\xi</code>	o	<code>o</code>
π	<code>\pi</code>	ρ	<code>\rho</code>	σ	<code>\sigma</code>	τ	<code>\tau</code>	v	<code>\upsilon</code>
ϕ	<code>\phi</code>	χ	<code>\chi</code>	ψ	<code>\psi</code>	ω	<code>\omega</code>		
Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>	Λ	<code>\Lambda</code>	Ξ	<code>\Xi</code>
Π	<code>\Pi</code>	Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>	Φ	<code>\Phi</code>	Ψ	<code>\Psi</code>
Ω	<code>\Omega</code>								
ε	<code>\varepsilon</code>	ϑ	<code>\vartheta</code>	ϖ	<code>\varpi</code>	ϱ	<code>\varrho</code>	ς	<code>\varsigma</code>
φ	<code>\varphi</code>								

4.2.2 演算記号

演算記号として用意されているもののうち、代表的なものをあげておく。

\pm	<code>\pm</code>	\mp	<code>\mp</code>	\times	<code>\times</code>	\div	<code>\div</code>		
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\ll	<code>\ll</code>	\gg	<code>\gg</code>	\neq	<code>\neq</code>
\equiv	<code>\equiv</code>	\approx	<code>\approx</code>	\cong	<code>\cong</code>	\doteq	<code>\doteq</code>	\propto	<code>\propto</code>
\cdot	<code>\cdot</code>	\circ	<code>\circ</code>	\bullet	<code>\bullet</code>				
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\otimes	<code>\otimes</code>	\oslash	<code>\oslash</code>	\odot	<code>\odot</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\cap	<code>\cap</code>	\cup	<code>\cup</code>	\aleph	<code>\aleph</code>
\subset	<code>\subset</code>	\subseteq	<code>\subseteq</code>	\supset	<code>\supset</code>	\supseteq	<code>\supseteq</code>	\emptyset	<code>\emptyset</code>
\wedge	<code>\wedge</code>	\vee	<code>\vee</code>	\neg	<code>\neg</code>	\forall	<code>\forall</code>	\exists	<code>\exists</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>	\top	<code>\top</code>	\perp	<code>\perp</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>		
\imath	<code>\imath</code>	\Re	<code>\Re</code>	\Im	<code>\Im</code>	∞	<code>\infty</code>	∇	<code>\nabla</code>

4.2.3 矢印類

矢印のたぐいで用意されているもののうち、代表的なものをあげておく。

\leftarrow	<code>\leftarrow</code>	\rightarrow	<code>\rightarrow</code>	\leftrightarrow	<code>\leftrightarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\longleftarrow	<code>\longleftarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Longleftarrow	<code>\Longleftarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\leftharpoondown	<code>\leftharpoondown</code>	\rightleftharpoons	<code>\rightleftharpoons</code>
\rightharpoonup	<code>\rightharpoonup</code>	\rightharpoondown	<code>\rightharpoondown</code>		
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>	\updownarrow	<code>\updownarrow</code>
\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>	\Updownarrow	<code>\Updownarrow</code>

4.2.4 初等関数

初等関数は、イタリック体ではなく、通常の字体で綴るのがふつうである。この目的のため、つぎのものが用意されている。

<code>\sin</code>	<code>\cos</code>	<code>\tan</code>	
<code>\sec</code>	<code>\csc</code>	<code>\cot</code>	
<code>\arcsin</code>	<code>\arccos</code>	<code>\arctan</code>	
<code>\sinh</code>	<code>\cosh</code>	<code>\tanh</code>	<code>\coth</code>
<code>\exp</code>	<code>\log</code>	<code>\ln</code>	<code>\lg</code>
<code>\arg</code>	<code>\deg</code>		
<code>\sup</code>	<code>\inf</code>	<code>\max</code>	<code>\min</code>
<code>\gcd</code>	<code>\det</code>	<code>\dim</code>	<code>\ker</code>

4.3 添字、分数など

添字や積分記号など、文字の大きさをかえ、書き出す位置も上下に動くものがある。

4.3.1 添字

上付の添字は `^` で、下付の添字は `_` で指示する。添字となるものは直後の 1 単位である。1 単位を超えるものを添字とするときは、`{ }` で括ること。

例 いくつかの例を示す。

<code>x^2</code>	<code>x^2</code>	<code>x^{2n}</code>	<code>x^{2n}</code>	<code>x^{2~n}</code>	<code>x^{2n}</code>
<code>x_2</code>	<code>x_2</code>	<code>x_{i j}</code>	<code>x_{ij}</code>	<code>x^{i j_n}</code>	<code>x_{ij_n}</code>
<code>x^2_i</code>	<code>x_i^2</code>	<code>x_i^2</code>	<code>x_i^2</code>	<code>x^{i_n}</code>	<code>x^{i_n}</code>

4.3.2 積分など

和や積、積分などの記号としては、つぎのものがある。その上下限などは、添字と同じ形で指示する。

<code>\sum</code>	\sum	<code>\prod</code>	\prod	<code>\coprod</code>	\coprod
<code>\int</code>	\int	<code>\bigcap</code>	\bigcap	<code>\bigcup</code>	\bigcup
<code>\bigsqcup</code>	\bigsqcup	<code>\bigwedge</code>	\bigwedge	<code>\bigvee</code>	\bigvee
<code>\lim</code>	\lim				

例 同じ式

`\lim_{n \rightarrow \infty} \sum_{i=0}^n x^n = \alpha`
 でも1行取りして書かせると
`\lim_{n \rightarrow \infty} \sum_{i=0}^n x^n = \alpha`
 となる。

⇒ 同じ式 $\lim_{n \rightarrow \infty} \sum_{i=0}^n x^n = \alpha$ でも1行取りして書かせると

$$\lim_{n \rightarrow \infty} \sum_{i=0}^n x^n = \alpha$$

となる。

4.3.3 分数

分数の形を指示するには、`\frac{ <分子> }{ <分母> }` と指示する。

例 `\frac{a+b+c}{2}` と書くと、つぎのように印字される。

$$\frac{a+b+c}{2}$$

4.3.4 根号

平方根を始めとする根号は、`\sqrt[n]{ <根号の中味> }` と指示する。 n が2(つまり平方根)のときは[2]全体を省略してよい。

例 `\(\sqrt{u+v+2\sqrt{uv}}=\sqrt{u}+\sqrt{v}\)` と指示すると、つぎのように印字される。

$$\sqrt{u+v+2\sqrt{uv}} = \sqrt{u} + \sqrt{v}$$

4.3.5 修飾

式の一部や変数にさまざまな修飾を付けることができる。

式 式の一部に修飾をつけるものとして、つぎのものがある。

<code>\overline{...}</code>	<code>\underline{...}</code>
<code>\overbrace{...}</code>	<code>\underbrace{...}</code>

例 それぞれの機能は、例を見るのが早い。

<code>\underline{\overline{x^2+1}}-y</code>	$\underline{\overline{x^2+1}}-y$
<code>\overbrace{a+b+\cdots+z}^{26}</code>	$\overbrace{a+b+\cdots+z}^{26}$

変数 変数に修飾をつけるもののいくつかを示す。

<code>\hat{x}</code>	\hat{x}	<code>\bar{x}</code>	\bar{x}	<code>\vec{x}</code>	\vec{x}
<code>\dot{x}</code>	\dot{x}	<code>\ddot{x}</code>	\ddot{x}		

積重ね 式を上下に積み重ねる。

`\stackrel{上式}{\underset{下式}{}}`

例 `\stackrel{a}{\longrightarrow}` ⇒ \xrightarrow{a}

4.3.6 省略記号

省略記号にはつぎのものがある。

`\ldots` ... `\cdots` ... `\vdots` ⋮ `\ddots` ⋱

例 たとえば、 x_1, \dots, x_n (`\ldots`) と $x_1 + \dots + x_n$ (`\cdots`) のように書き分ける。

4.3.7 法

整数 n を法とする合同を扱うのにつぎの二つの書き方がある。

演算 `\bmod` は演算記号となる。

例 $x = a \bmod n$ は、“ $x = a \bmod n$ ” と印字される。

関係 `\pmod{ <法> }` は、合同関係を示すのに使う。

例 $x^2 \equiv 1 \pmod{n}$ は、“ $x^2 \equiv 1 \pmod{n}$ ” と印字される。

4.4 行列

行列の中味を書くには、つぎの形に書く。

```
\begin{array}{<列指定>}
{ <列項目> { & <列項目> } \\
}
<列項目> { & <列項目> }
\end{array}
```

<列指定> の書き方は、`tabular`(3.5.2 参照) と同じであるが罫線の指定はできない。

例 $\begin{array}{ccc} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{array}$ \implies $\begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

行列の中味をくくる括弧は、その中味に応じた大きさになってほしい。このような機能を持つ記号には、つぎのものがある。

(()) [[]]
{ \{ } \} [\lfloor] \rfloor
[\lceil] \rceil < \langle \rangle \rangle
| | || \|\

これらの記号を、行列の中味に応じた大きさにするには、`\left` と `\right` を前置して使う。

例 たとえば、 $\left(\begin{array}{ccc} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{array} \right)$ と先程の行列の中味を囲ってやると、つぎのように印字される。

$$\left(\begin{array}{ccc} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{array} \right)$$

例 場合分けをした定義などもこの応用で書くことができる。

$f_n = \left\{ \begin{array}{ll} 0 & (n=0) \\ 1 & (n=1) \\ f_{n-1} + f_{n-2} & (n>1) \end{array} \right.$

上の例で出てきたように、`\right.` と指定すると、空の記号として扱われる。

4.5 複数行の数式

等式の変形を複数行にわたって行ったり、複数の等式を表示したりするには、つぎの書き方がある。

```
\begin{eqnarray} < 等式の列 > \end{eqnarray}
\begin{eqnarray*} < 等式の列 > \end{eqnarray*}
```

両者の違いは、それぞれの等式に一連の番号を付けるかどうかの違いである。

等式の列は、つぎの形を \\ で区切って並べて書く。

< 左辺 > & < 等号など > & < 右辺 >

& の位置が上下の行で揃うのは \tabular(3.5.2 参照) や \array(4.4 参照) と同じである。

なお、\eqnarray の中で番号を振りたくない行に対しては、その行の終わりに \nonumber と指定しておく。

例 つぎのように書くと

```
\begin{eqnarray}
\sin(x+y) &=& \sin x \cos y + \cos x \sin y \nonumber \\
\cos(x+y) &=& \cos x \cos y - \sin x \sin y
\end{eqnarray}
```

たとえば、つぎのように印字される。

$$\begin{aligned} \sin(x+y) &= \sin x \cos y + \cos x \sin y \\ \cos(x+y) &= \cos x \cos y - \sin x \sin y \end{aligned} \quad (1)$$

また、\lefteqn{...} で括ると、その部分の大きさは 0 文字として扱われるので、等式の変形などで最初の行をこれで括っておけば、見やすく配置することができる。

```
例 \begin{eqnarray*}
\lefteqn{x^2-xy+y^2} && \Rightarrow && x^2 - xy + y^2 \\
&& && = (x - \frac{y}{2})^2 + \frac{3}{4}y^2 \\
&& && \geq 0 \\
&& && \text{\&\geq\& 0} \\
\end{eqnarray*}
```

4.6 ヒント

数式の中に、通常の文章と同じような印字になる部分を混在させたいことがある。このときは、その部分を \mbox{...} と括ってやるとよい。

```
例 \[ f(x) = \left\{ \begin{array}{ll} x^3 & \text{\mbox{(if \$x\geq 0\$)}} \\ 0 & \text{\mbox{(otherwise)}} \end{array} \right. \Rightarrow f(x) = \begin{cases} x^3 & (\text{if } x \geq 0) \\ 0 & (\text{otherwise}) \end{cases}
```

5 定義

L^AT_EX が用意してしてくれる指示に加えて、自分なりの新しい指示の書き方を定義して使うこともできる。

字体の指示には、つぎのものがある。

```
イタリック \it      ゴシック      \gt
ボールド   \bf      タイプライタ  \tt
```

字体の変更は、それ以後の文面 ({ と } で括られている部分、または \begin{...} と \end{...} で括られている部分) の全体にわたる。したがって、特定の部分だけ字体を変えるには、その部分を { と } で括って、たとえば、{\bf ...} のようにする。

ところで、字体を変更したい、というのは、印字結果に対する希望であって、そうしなくなった文書上での理由は一通りではない。単なる強調をしたいこともあれば、初出の用語を示したいこともある。L^AT_EX 文書は、できるだけ、こうした文書上での“構造”を明示するように書くのがよい。

そこで、初出の用語を示す、という意味の指示 \first を用意して、\first{< 初出用語 >} と指示できるようにしよう。

```
\newcommand{\first}[1]{\bf #1}
```

この定義は、本体の前の < 定義 > の位置に書く。{\first} は、この指示の名前を定義する。[1] は、引数の個数を定義する。そして最後の { と } の中にその指示を使ったときに置き換えるべき文面を書く。このとき、#1 は、実際の第 1 引数で置き換えられる。

同様に、\begin{...} と \end{...} とで囲って使う形の指示を定義することもできる。たとえば、算譜などを引用する場合、行間を詰めた表示にしたい。行間は、\baselineskip という名前の変数に収められている。このような変数の値を変更するには、つぎの形で指示する。

```
\setlength{変数名}{新しい値}
```

この指示も、それ以後の文面にわたる。そこで、たとえば、つぎのような定義を与えて使うことにするのである。

```
\newenvironment{program}%  
{\begin{quote}\setlength{\baselineskip}{0.8 \baselineskip}}%  
{\end{quote}}
```

今度は、{ と } で括ったものが三つ必要になる。最初のものは、名前を定義する。次のものは、\begin{...} に置き換わるべき文面を定義する。第 3 のものは、\end{...} に置き換わるべき文面を定義する。これらは、改行なしで書き並べるのが原則であるので、上の例では行末に % を置いて改行を無効にしている。

6 図・表など

図や表は、ページをまたがるわけにいかない。そこで、 \LaTeX に任せて、適当な位置に配置してもらおうようにするとよい。このような要素を、浮動要素という。

6.1 浮動要素

浮動要素と指定するには、つぎの形に書く。

```
\begin{figure} < 図本体 > \end{figure}  
\begin{table} < 表本体 > \end{table}
```

それぞれの本体には、図や表を書くほか、その説明を書いてもよい。特に、見出しを付けることができる。それには、つぎの形で指定する。

```
\caption{ < 見出し > }
```

見出しは、図であれば“図 n ”と、表であれば“表 n ”とそれぞれの通し番号が振られる。この番号を文章中で参照するには、\label{ < 参照略号 > } を \caption{ ... } の直後に書き、参照する箇所に \ref{ < 参照略号 > } と書けばよい。参照略号については、2.2 を参照のこと。

表そのものは、\begin{tabular} ... \end{tabular} で作成するのがふつうである。

一方、図は、適切な道具を使って作成することが多い。たとえば、Mathematica で図を作成する。こうした道具で作成した図は、EPS(Extended PostScript) 形式で独立した算帖 (file) にしまっておく。その図を引用するには、つぎの形を使う。

```
\epsfile{file= 算帖名 ,scale= 倍率 }
```

ここで、倍率は、作成したときの図に対してどれぐらいの倍率で取り込むのかを指定する。作成したときの大きさのままであれば、1.0 と指定する。

このとき、忘れずに文書の頭を書く \documentstyle の < 書式追加指定 > に eclpsf を書き加えておく (1.1 参照)。

6.2 区画

浮動要素ではないが、ページの中に、あたかも小さな別のページがあるかのような取り扱いをしたいことがある。そうしたときには、つぎの形でその小ページを指定する。

```
\begin{minipage}[ < 配置 > ]{ n mm}  
  < 本体 >  
\end{minipage}
```

小ページの横の大きさ(幅)は、 n で指定する。縦の大きさは、< 本体 > に書いたものから自動的に決まる。

小ページ全体は、それ全体が(大きな)1文字であるとして扱われる。このとき、その全体の上端(t)で位置合せをするか、下端(b)で位置合せをするか、を< 配置 > に指定する。[と]を含めて< 配置 > を省けば、中央で位置合せを行う。