

# 第1章 数式

この章では数式を扱いたいと思います。数式を打つために  $\text{T}_{\text{E}}\text{X}$  を使い始める人が大半だと思いますので、頑張っていきましょう。

## 1.1 数式の基礎

数式は、 $\$$ または $\backslash[ \ ]$ で挟んだ部分に書きます。文章中に数式を入れたいときは、 $\$$ で挟めばよいです。例えば  $y = ax + b$  こんな風に ( $\$y = ax + b\$$ )。

重要な数式は文章とは別で1行使いたいですね。そんなときは $\backslash[ \ ]$ で挟んで、

```
\[ y = ax + b \]
```

のようにすれば、

$$y = ax + b$$

のようになります。

上の例を見ていただけるとわかると思うのですが、数式中に半角スペース（や `tab`）を入れても出力されません（つまり、入れないのと同じです）。見やすくするために適宜半角スペースを入れるのがよいと思います。

また、数式は、AMS（米国数学会）が開発した「`amsmath`, `amssymb`」というパッケージで拡張することができます。とりあえずプリアンブル（`\documentclass` と `\begin{document}` の間）に

```
\usepackage{amsmath, amssymb}
```

と書いてください。

## 1.2 上付き文字と下付き文字

指数、例えば  $x^2$  を出力するには、「 $\wedge$ 」を用いて、

```
$ x^2 $
```

とすればよいです。この、「 $^2$ 」のような、文字などの右肩につく文字を「上付き文字」といいます。上付き文字が複数になるときは、`{}`でくくって

```
$ x^{\{n-1\}} $  $\rightarrow x^{n-1}$ 
```

とします。くくらないと変なことになったりするので気を付けて下さい。

$a_n$  のように文字の右下につく文字を「下付き文字」と言います。下付き文字を出力するには、「 $_$ 」を用いて、

```
$ a_n $  $\rightarrow a_n$ 
```

とすればよいです。上付き文字と同様、下付き文字が複数になるときは

`$ a_{n-1} $`  $\rightarrow a_{n-1}$

のようにすればよいです。

下にいくつか例を挙げておきます。なお、`\mathrm` は、デフォルトではイタリック体になってしまう数式中の文字をローマン体にするコマンドです。

<code>a_{ij}</code>	$\rightarrow a_{ij}$
<code>{v_0}^2</code>	$\rightarrow v_0^2$
<code>\mathrm{{}^1{}_1H}</code>	$\rightarrow {}^1_1\mathrm{H}$
<code>a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0</code>	$\rightarrow a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$

## 1.3 演算記号

次は演算記号についてです。`+`, `-`, `=` はキーボードから打ったものをそのまま使えますが、他の色々な記号はコマンド (`\` で始まるもの) を使って出力します。例えば、掛け算記号の  $\times$  は、`\times` というコマンドを使えば出力できます。また、掛け算やベクトルの内積を表す「 $\cdot$ 」は、`\cdot` で出せます。

こういったものを下にまとめておきます。ただ、あまり充実させてはいないので、ここに載っていないものは是非自分で調べてみて下さい。

入力	出力		入力	出力		入力	出力
<code>+</code>	$+$		<code>&lt;</code>	$<$			
<code>-</code>	$-$		<code>\leq</code>	$\leq$		<code>\prime</code>	$'$
<code>\pm</code>	$\pm$	入力	<code>\ll</code>	$\ll$		<code>\emptyset</code>	$\emptyset$
<code>\mp</code>	$\mp$	<code>=</code>	<code>&gt;</code>	$>$		<code>\nabla</code>	$\nabla$
<code>\times</code>	$\times$	<code>\equiv</code>	<code>\geq</code>	$\geq$		<code>\angle</code>	$\angle$
<code>\div</code>	$\div$	<code>\sim</code>	<code>\gg</code>	$\gg$		<code>\triangle</code>	$\triangle$
<code>*</code>	$*$	<code>\simeq</code>	<code>\subset</code>	$\subset$		<code>\forall</code>	$\forall$
<code>\circ</code>	$\circ$	<code>\approx</code>	<code>\subseteq</code>	$\subseteq$		<code>\exists</code>	$\exists$
<code>\bullet</code>	$\bullet$	<code>\propto</code>	<code>\supset</code>	$\supset$		<code>\neg</code>	$\neg$
<code>\cdot</code>	$\cdot$	<code>\perp</code>	<code>\supseteq</code>	$\supseteq$			
<code>\cap</code>	$\cap$	<code>\parallel</code>	<code>\in</code>	$\in$			
<code>\cup</code>	$\cup$		<code>\ni</code>	$\ni$			
<code>\vee</code>	$\vee$						
<code>\wedge</code>	$\wedge$						

`amsmath` パッケージを読み込むともっと多くの記号を使えるようになります。こちらもとりにあえず使えそうなものだけ挙げておきます。

入力	出力
<code>\therefore</code>	$\therefore$
<code>\because</code>	$\because$
<code>\leqq</code>	$\leqq$
<code>\geqq</code>	$\geqq$
<code>\fallingdotseq</code>	$\fallingdotseq$

## 1.4 分数

分数は`\frac` というコマンドを使って出力します。使い方は、

```
\frac{分子}{分母}
```

です。

<code>\[ \frac{1}{2} \]</code>	$\frac{1}{2}$
<code>\[ \frac{13}{100} \]</code>	$\frac{13}{100}$
<code>\[ \frac{1}{1 + x^2} \]</code>	$\frac{1}{1+x^2}$

## 1.5 平方根

平方根は`\sqrt` というコマンドで出力できます。使い方は、

```
\sqrt{ルートの中身}
```

です。

<code>\[ \sqrt{2} \]</code>	$\sqrt{2}$
<code>\[ \sqrt{10} \]</code>	$\sqrt{10}$
<code>\[ \sqrt{b^2 - 4ac} \]</code>	$\sqrt{b^2 - 4ac}$
<code>\[ \sqrt{\frac{1}{2}} \]</code>	$\sqrt{\frac{1}{2}}$

## 1.6 括弧

普通に使う括弧としては

```
() , {} , [] , ||
```

の 4 つくらいだと思います。() と [] と || はそのまま打てば出せます。ただ、{} は  $\text{T}_{\text{E}}\text{X}$  では特殊な働きをするので、そのまま打っても出力されません。{} を出すには、\をつけて\{ \}とします。

普通に () をつけただけでは、中に分数を入れたときなどにサイズが合わず不格好になってしまいます。こういうときのために`\bigl`などのコマンドが用意されていますが、基本は`\left`と`\right`を付けて

```
\left( \right)
```

などとします。

```
\[ (x - 1)(x - 5) = x^2 - 6x + 5 \]
\[ n \left( \frac{3}{2} R \right) T \]
\[ \left\{ \left( x - \frac{3}{5} \right) + \frac{1}{2} \right\}
```

$$(x-1)(x-5) = x^2 - 6x + 5$$

$$n \left( \frac{3}{2} R \right) T$$

$$\left\{ \left( x - \frac{3}{5} \right) + \frac{1}{2} \right\}$$

## 1.7 ギリシャ文字

ギリシャ文字はコマンドを使って出力します。小文字は英語名 (alpha など) をそのままコマンドにしたもの (つまり \ をつけたもの) で出力できます。大文字は小文字のスペルの 1 文字目を大文字にするだけです。また,  $\omicron$  (omicron) は英語の  $\omicron$  (オー) のイタリック体  $\omicron$  と同じなので用意されていません。とりあえず下にまとめておきます。

入力	出力	入力	出力	入力	出力	入力	出力
\alpha	$\alpha$	\eta	$\eta$	\nu	$\nu$	\tau	$\tau$
\beta	$\beta$	\theta	$\theta$	\xi	$\xi$	\upsilon	$\upsilon$
\gamma	$\gamma$	\iota	$\iota$	\textit{o}	$\textit{o}$	\phi	$\phi$
\delta	$\delta$	\kappa	$\kappa$	\pi	$\pi$	\chi	$\chi$
\epsilon	$\epsilon$	\lambda	$\lambda$	\rho	$\rho$	\psi	$\psi$
\zeta	$\zeta$	\mu	$\mu$	\sigma	$\sigma$	\omega	$\omega$

一部の文字には変体文字が用意されています。

入力	出力	入力	出力	入力	出力
\varepsilon	$\epsilon$	\varphi	$\phi$	\varsigma	$\varsigma$
\vartheta	$\vartheta$	\varrho	$\rho$	\varphi	$\phi$

大文字は以下の 11 個以外はアルファベットと同じです。

入力	出力	入力	出力	入力	出力
\Gamma	$\Gamma$	\Lambda	$\Lambda$	\Psi	$\Psi$
\Delta	$\Delta$	\Xi	$\Xi$	\Omega	$\Omega$
\Theta	$\Theta$	\Pi	$\Pi$		

## 1.8 矢印

矢印も色々出せます。

入力	出力	入力	出力
<code>\uparrow</code>	$\uparrow$	<code>\rightarrow(\to)</code>	$\rightarrow$
<code>\Uparrow</code>	$\Uparrow$	<code>\Rightarrow</code>	$\Rightarrow$
<code>\downarrow</code>	$\downarrow$	<code>\leftarrow(\gets)</code>	$\leftarrow$
<code>\Downarrow</code>	$\Downarrow$	<code>\Leftarrow</code>	$\Leftarrow$
<code>\updownarrow</code>	$\updownarrow$	<code>\leftrightarrow</code>	$\leftrightarrow$
<code>\Updownarrow</code>	$\Updownarrow$	<code>\Leftrightarrow</code>	$\Leftrightarrow$
入力	出力	入力	出力
<code>\longrightarrow</code>	$\longrightarrow$	<code>\mapsto</code>	$\mapsto$
<code>\Longrightarrow</code>	$\Longrightarrow$	<code>\longmapsto</code>	$\longmapsto$
<code>\longleftarrow</code>	$\longleftarrow$	<code>\rightharpoonup</code>	$\rightharpoonup$
<code>\Longleftarrow</code>	$\Longleftarrow$	<code>\rightharpoondown</code>	$\rightharpoondown$
<code>\longlefttrightarrow</code>	$\longleftrightarrow$	<code>\nearrow</code>	$\nearrow$
<code>\Longlefttrightarrow</code>	$\Longleftrightarrow$	<code>\searrow</code>	$\searrow$

## 1.9 式の上下に付けるもの

数式の上下には、矢印や線、点といった色々なものを付けることができます。

入力	出力
<code>\bar{x}</code>	$\bar{x}$
<code>\tilde{x}</code>	$\tilde{x}$
<code>\vec{x}</code>	$\vec{x}$
<code>\dot{x}</code>	$\dot{x}$
<code>\ddot{x}</code>	$\ddot{x}$

また、伸縮するものもあります。

入力	出力	入力	出力
<code>\overline{a + b}</code>	$\overline{a+b}$	<code>\overrightarrow{a + b}</code>	$\overrightarrow{a+b}$
<code>\underline{a + b}</code>	$\underline{a+b}$	<code>\overleftarrow{a + b}</code>	$\overleftarrow{a+b}$
<code>\widehat{abc}</code>	$\widehat{abc}$	<code>\overbrace{abc}</code>	$\overbrace{abc}$
<code>\widetilde{abc}</code>	$\widetilde{abc}$	<code>\underbrace{abc}</code>	$\underbrace{abc}$

`\overbrace` と `\underbrace` については、以下のような使い方もできます。

```
\[
{}_m \mathrm{P}_n =
\overbrace{m \cdot (m-1) \cdot \dots (m-n+1)}^{n \text{ コ}}
\]
\[
{}_m \mathrm{P}_n =
\underbrace{m \cdot (m-1) \cdot \dots (m-n+1)}_{n \text{ コ}}
\]
```

$${}_m P_n = \overbrace{m \cdot (m-1) \cdot \dots (m-n+1)}^{n \text{ コ}}$$

$${}_m P_n = \underbrace{m \cdot (m-1) \cdot \dots (m-n+1)}_{n \text{ コ}}$$

また、矢印の上に文字を書きたいときもあると思います。そのときは、

```
\stackrel{数式}{\rightarrow}
```

を使えばよいです。

```
\[
x \stackrel{f}{\rightarrow} y
\]
U \stackrel{\mathrm{def}}{=} n C_V T
\]
```

$$x \xrightarrow{f} y$$

$$U \stackrel{\mathrm{def}}{=} n C_V T$$

## 1.10 数式の書体とフォント

第??章??節で説明していますが、数式の中にはイタリック体にしないものもあります。しかし、数式モード中では自動でイタリック体になってしまいます。これをアップライト体（立体）に戻すために、`\mathrm`というコマンドが用意されています。また、`amsmath` パッケージの`\text` コマンドを使えば数式中にテキストを入れることができます。

```
\[
x_{\mathrm{max}}
\]
\[
x + \mathrm{Const.}
\]
\[
x \text{ , } \mathrm{cm}^2
\]
\[
v_{\text{水平}}
\]
\[
f(x) = x^2 \text{ (二次関数)}
\]
```

$$x_{\mathrm{max}}$$

$$x + \mathrm{Const.}$$

$$x \text{ , } \mathrm{cm}^2$$

$$v_{\text{水平}}$$

$$f(x) = x^2 \text{ (二次関数)}$$

他にも色々なフォント・字体があるので、一気に下の表にまとめてしまいます。

- 立体 : `\mathrm`
- 筆記体 : `\mathcal` `eucal` パッケージの使用で形状変化
- 太字 : `\bm` `bm` パッケージの使用が必要
- RSFS : `\mathsrc` `mathrsfs` パッケージの使用が必要
- RSFS : `\mathcal` `rsrso` パッケージの使用で、筆記体がマイルドな RSFS 体に
- ドイツ文字 : `\mathfrak`
- 黒板文字 : `\mathbb`

下に例を示します。筆記体ははみ出しました。ごめんなさい。

```

\begin{align*}
&\&
\mathrm{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\
&\&
ABCDEFGHIJKLMNOPQRSTUVWXYZ \\
&\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\
&\&
\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\
&\&
\bm{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\
&\&
\bm{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\
&\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\
&\&
\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\
&\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\
&\&
\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\
&\end{align*}

```

`bm` パッケージについてですが、`nextmath` や `mathpazo` 等の数式フォントを設定するパッケージを使用している場合はそれらの後に読み込むようにしてください。

## 1.11 数学関数

`sin` や `cos`, `log` といった関数は、イタリック体ではなく立体で書くことになっています (第??章??節)。これを書くために、それぞれの関数に`\`をつけた命令が用意されています。気を付けてみて下さい。

$\times \sin x$     $\bigcirc \sin x$

## 1.12 数式番号と参照

いままで、別行立ての数式は`\[ \]`で出力する、と言っていましたが、他にも色々方法はあります。例えば、`equation` 環境というものがあります。これは、`\[ \]` のように別行立ての数式を出力し、更に数式番号を自動で付けてくれます。下の例では、第 1 章の 1 つめの数式ということで、(4.1) という番号が付けられています。

```

\begin{equation}
y = ax + b
\end{equation}

```

$$y = ax + b \quad (1.1)$$

```

\begin{equation*}
y = cx + d
\end{equation*}

```

$$y = cx + d$$

もし数式番号を付けたくないのために、`equation*`環境というものが用意されています。これはほぼ `\[ \]` と同じものになります。

数字でない、例えば `*` のような番号を付けたいときは、`\tag` というコマンドを使って、

```

\begin{equation*}
v = v_0 + at \tag{$*$}
\end{equation*}

```

$$v = v_0 + at \quad (*)$$

のようにします。() が不要なときは `\tag*` というコマンドを使えば OK です。

$\text{\TeX}$  では、数式の参照を自動で行うことができます。数式の参照とは、「式 (4.1) において...」などのようなものです。これは、数式にラベルを張り、それを参照する、という形で行います。使い方ですが、

```

\begin{equation}
y = ax^2 + bx + c
\label{eq:parabola}
\end{equation}

```

$$y = ax^2 + bx + c \quad (1.2)$$

```

\pageref{eq:parabola}ページの
式~(\ref{eq:parabola}) におい
て,$a=0$のとき...

```

$$8 \text{ ページの式 (1.2) において, } a = 0 \text{ のとき...}$$

のようにします。参照したい式に `\label` コマンドでラベルを貼り、後に使いたいところで `\ref` を用いてその式番号を出力することができます。また、その式のあるページ数を知りたいのために、`\pageref` というコマンドも用意されているので、必要に応じて使ってみてください。また、式の参照には `\eqref` というコマンドが用意されています。これはキレイに参照番号等を出してくれるので、数式の参照においてはこちらを使ってもいいと思います。

なお、第??章??節で説明している通りですが、2 回実行しないと正しく参照が表示されないなので、そこは注意してください。



## 1.13 数式の頭揃え

式変形を書くときは、数行に渡って = などの位置で揃えて数式を書きたいときもあります。例えば下の様なものです。

$$\begin{aligned}
 I &= \int_0^{\infty} \frac{1}{1+x^2} dx \\
 &= \int_0^{\frac{\pi}{2}} \frac{1}{1+\tan^2 \theta} \cdot \frac{1}{\cos^2 \theta} d\theta \\
 &= \int_0^{\frac{\pi}{2}} 1 d\theta = [\theta]_0^{\frac{\pi}{2}} \\
 &= \frac{\pi}{2}
 \end{aligned}$$

これを出力するには、`align` 環境を使います。この環境は `amsmath` パッケージで定義されているので、まずはプリアンブル (`\documentclass` と `\begin{document}` の間) に

```
\usepackage{amsmath}
```

と書いてください。

使い方ですが、揃える位置に `&` を書き、改行する位置に `\\` を書けば OK です。デフォルトでは各行に式番号がついてしましますが、不要なところには `\notag` を付ければその行にはつきません。

```

\begin{align}
I \quad & \& = \int_0^{\infty} \frac{1}{1+x^2} dx \\
& \& = \int_0^{\frac{\pi}{2}} \frac{1}{1+\tan^2 \theta} \cdot \frac{1}{\cos^2 \theta} d\theta \\
& \& = \int_0^{\frac{\pi}{2}} 1 d\theta = [\theta]_0^{\frac{\pi}{2}} \\
& \& = \frac{\pi}{2}
\end{align}

```

$$I = \int_0^{\infty} \frac{1}{1+x^2} dx \tag{1.3}$$

$$= \int_0^{\frac{\pi}{2}} \frac{1}{1+\tan^2 \theta} \cdot \frac{1}{\cos^2 \theta} d\theta \tag{1.4}$$

$$= \int_0^{\frac{\pi}{2}} 1 d\theta = [\theta]_0^{\frac{\pi}{2}} = \frac{\pi}{2} \tag{1.5}$$

数式番号が不要ならば `align*` 環境を用いて下さい。また、`align` 環境は表と同じように何列でも揃えられるようになっています。例えば、次のように複数の数式を揃えられるということです。

```
\begin{align*}
x &= r \cos\theta \cos\varphi, & y &= r \cos\theta \sin\varphi, & z &= r \\
&\sin\theta \\
r &= a, & \theta &= \frac{\pi}{2}, & \varphi &= \frac{\pi}{3} \\
\end{align*}
```

$$\begin{aligned} x &= r \cos \theta \cos \varphi, & y &= r \cos \theta \sin \varphi, & z &= r \sin \theta \\ r &= a, & \theta &= \frac{\pi}{2}, & \varphi &= \frac{\pi}{3} \end{aligned}$$

&で揃え、\\で改行するのは、第??章で扱う表と全く同じなので、 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  に慣れてくれば自然と行えるようになると思います。

align 環境では数式各行に式番号が振られますが、複数行の数式全体に 1 つの番号を振りたいときは、equation 環境と split 環境を使って、

```
\begin{equation}
\begin{split}
e^x &= \sum_{n=0}^{\infty} \frac{1}{n!} x^n \\
&= 1 + x + \frac{1}{2!} x^2 \\
&+ \frac{1}{3!} x^3 + \dots
\end{split}
\end{equation}
```

$$\begin{aligned} e^x &= \sum_{n=0}^{\infty} \frac{1}{n!} x^n \\ &= 1 + x + \frac{1}{2!} x^2 + \frac{1}{3!} x^3 + \dots \end{aligned} \quad (1.6)$$

他にも、\fbox を使うときに便利な aligned 環境、数式間の空白を自分で制御できる alignat 環境、1 つの数式が複数行に渡るときに使える multiline 環境などがあります。また、align 環境の途中で一旦テキスト（すなわち、一般に、など）を入れてからもう一度数式に戻りたいときには\intertext{テキスト}というコマンドが便利です。ここに書いてあることで足りないときは是非調べてみてください。

## 1.14 場合分け

場合分けは cases 環境を用いて出力できます。

```
\[
| x | =
\begin{cases}
x & (x \leq 0) \\
-x & (x < 0)
\end{cases}
\]
```

$$|x| = \begin{cases} x & (x \leq 0) \\ -x & (x < 0) \end{cases}$$

## 1.15 行列

amsmath パッケージで、数種類の行列用の環境が定義されています。

```

\[
\begin{matrix}
a & b \\
c & d
\end{matrix}
\\
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix}
\\
\begin{bmatrix}
a & b \\
c & d
\end{bmatrix}
\\
\begin{Bmatrix}
a & b \\
c & d
\end{Bmatrix}
\\
\begin{vmatrix}
a & b \\
c & d
\end{vmatrix}
\\
\begin{Vmatrix}
a & b \\
c & d
\end{Vmatrix}

```

$$\begin{matrix} a & b \\ c & d \end{matrix} \\
 \begin{pmatrix} a & b \\ c & d \end{pmatrix} \\
 \begin{bmatrix} a & b \\ c & d \end{bmatrix} \\
 \begin{Bmatrix} a & b \\ c & d \end{Bmatrix} \\
 \begin{vmatrix} a & b \\ c & d \end{vmatrix} \\
 \begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

表や `align` 環境と同じで、列の区切りを`&`、行の区切りを`\\`で行います。  
また、

```
\hdotsfor{列数}
```

というコマンドで、複数列に渡る点々を書くことができます。

```

\[
A =
\begin{pmatrix}
a_{11} & \hdots & a_{1n} \\
\hdotsfor{3} & & \\
a_{m1} & \hdots & a_{mn}
\end{pmatrix}

```

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ & \dots & \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$$