

TeX で色々な数式を記述する

栗野 俊一*

Ver. 0.03a (2012/06/12 版)

1 TeX の基本

1.1 文章

TeX で文章を作成する場合、まずは、base.tex の「本文」の部分に普通に日本語の文字列を入力します。すると、その入力された文章が、そのまま整形されて表示されるようになります。

非整形文章

例えば、\TeX の文中にこんなふうに、いい加減に、
文章の途
中に改行を入れてみます。
そうやって、長い文を、複数の行に分けて記入します。
もちろん、一行の中に複数の文を入れてもかまいませんが、
それはとっても編集しにくい文章になってしまうわけです。

これは、\TeX の良いところです。
なぜなら、「表示に都合が良い形が、必ずしも編集に都合が良いとは限らない」からです。
\TeX では「編集しやすい形で内容を編集」し、
それを自動的に「表示に都合が良い形に変換して表示」してくれます。
だから、\TeX の利用者は、
「あまり表示結果の事を気にせず、内容の作成に専念する」
事ができます。

これは、TeX で変換すると次のように整形され、右揃えして表示されます。また、空行があると、段落の区切と解釈され、段落をわけて^{*1}くれます。

* 日本大学理工学部数学科 講師 (kurino@math.cst.nihon-u.ac.jp)

*1 もちろん、日本語の習慣にあわせて、段落の頭は、一文字下げもしてくれます。

整形後の文章

例えば、 \TeX の文中にこんなふうに、いい加減に、文章の途中に改行を入れてみます。そうやって、長い文を、複数の行に分けて記入します。もちろん、一行の中に複数の文を入れてもかまいませんが、それはとても編集しにくい文章になってしまうわけです。

これは、 \TeX の良いところです。なぜなら、「表示に都合が良い形が、必ずしも編集に都合が良いとは限らない」からです。 \TeX では「編集しやすい形で内容を編集」し、それを自動的に「表示に都合が良い形に変換して表示」してくれます。だから、 \TeX の利用者は、「あまり表示結果の事を気にせず、内容の作成に専念する」事ができます。

1.2 マクロ

特別な表示（例えば、数式）を行いたい場合は、通常、「 \backslash 」（バックスラッシュ、あるいは円マーク「 \yen 」）の後に英字列を並べたもの（例えば、「 $\backslash\text{\TeX}$ 」や、「 $\yen\text{\TeX}$ 」など、これは「 \TeX 」と表示される）を記入します。

これらは、 \TeX にとって、特別な意味があり、これを他の文字列と区別して、「マクロ」と呼んでいます*2。

\TeX を学ぶ事の一つは、どのようなマクロがあり、また、それがどのような影響を持っているかを知る事になります。

1.3 注意：バックスラッシュと円マーク

テキストや、Web 上での表記が「 \backslash 」なのに、普段利用する時には「 \yen 」を使うのは「日本の特殊事情*3」だと思ってください。

以下、この文章は、「 \backslash 」を使いますが、皆さんは、すべてこれを「 \yen 」に置き換えて考えてください。

2 数式の基本

\TeX の中で数式を利用する場合、その数式が文中にあるか、それとも段落として独立しているかを区別する必要があります。

2.1 数式を文中で利用する場合

数式を文中で利用する場合は、数式を「 $\$$ 」で挟む必要があります。例えば、次のようにします。

文中に式を入れる例 (\TeX)

文中に数式 $y=\frac{1}{x}$ を入れた場合。

*2 これをなぜ、「マクロ」と呼ぶのかは、後日説明します。

*3 コンピュータでは、コード（文字を表現する数値）に対してどの文字（の表示の形 [フォント]）を表示するかを表（文字コード表）を定め、その表に基いて、文字の表示を行っている。

海外では、ASCII コード表（主に、半角文字に対応する数値の表で、数値 $0x5C$ には \backslash [バックスラッシュ] を対応させる）が利用されるが日本では JIS X 0201 コード表（ASCII の範囲では、 $0x5C$ 以外は同じフォントだが、 $0x5C$ だけ \yen [円マーク] に対応させる）が利用されているため。

ここで、`\frac`は分数を表現するマクロです。これは、次のように表示されます。

文中に式を入れる例 (タイプセット後) —————

文中に数式 $y = \frac{1}{x}$ を入れた場合。

2.2 数式を段落として独立させる場合

数式を段落として独立させる場合は「`\[`」と「`\]`」で挟みます。

段落としての数式 (T_EX) —————

段落として数式

`\[`

`y=\frac{1}{x}`

`\]`

を入れる場合。

これは、次のように表示されます。

段落としての数式 (タイプセット後) —————

段落として数式

$$y = \frac{1}{x}$$

を入れる場合。

このように、`$` と `$` や、`\[` と `\]` に挟まれた部分は、「数式モードになっている」と言います。

2.3 文中でも数式を大きく表示させたい場合

文中の数式と、段落の場合の数式では、色々表示が異なります。上記の例のように分数や、総和などを利用する場合は、大きさや位置が変化します。

文中の数式 (T_EX) —

文中に分数 $y=\frac{1}{x}$ や、
総和 $\sum_{i=1}^n(2\times i-1) = n^2$ を入れると
不格好です。

次のように段落として、記述すれば、綺麗なのですが。

```
\[  
y=\frac{1}{x}  
\]
```

```
\[  
\sum_{i=1}^n(2\times i-1) = n^2  
\]
```

ここで、`\sum` は総和を、`\times` は、積を表すマクロです。これは、次のように表示されます。

文中の数式 (タイプセット後) —

文中に分数 $y = \frac{1}{x}$ や、総和 $\sum_{i=1}^n(2 \times i - 1) = n^2$ を入れると不格好です。
次のように段落として、記述すれば、綺麗なのですが。

$$y = \frac{1}{x}$$

$$\sum_{i=1}^n(2 \times i - 1) = n^2$$

このような場合は、`\displaystyle` というマクロを先行させれば綺麗な表記になります。ただし、行間が空くので、数式が綺麗でも、ページとしてはちょっと不格好になります。

`displaystyle` の利用 (T_EX) —

文中の数式でも $\displaystyle\frac{1}{x}$ のようにできます。

これは、次のように表示されます。

`displaystyle` の利用 (タイプセット後) —

文中の数式でも $\frac{1}{x}$ のようにできます。

3 色々な数式

3.1 数や変数

数字列からなる数値や、英字 1 文字からなる変数^{*4} は、入力したものが、そのまま表示されます。ただし、フォントがイタリックに変化し、数式内の変数や数値である (数式モードである) 事が解ります。

数式内の英字 (T_EX) —

ふつうに「123」や「xyz」とした場合と数式モードにした「\$123\$」や「\$xyz\$」では微妙に表示が異なります。

数式内の英字 (タイプセット後) —

ふつうに「123」や「xyz」とした場合と数式モードにした「123」や「xyz」では微妙に表示が異なります。

3.2 指数や添字

指数や、添字を付ける場合は、「^」や「_」を使います。

上付き、下付き文字 (T_EX) —

指数は、 x^2 のように「^」を利用し、添字は x_i のように「_」を利用します。

上付、下付文字 (タイプセット後) —

指数は、 x^2 のように「^」を利用し、添字は x_i のように「_」を利用します。

3.3 範囲指定

上記の「^」や「_」は、後にくる「一つのもの」だけを、指数や、添字として扱います。例えば、良くある失敗は、次のようなものです。

上付き、下付き文字の範囲の指定の失敗例 —

x^{10} とすると x^{10} になってしまい、 x^{10} とならない。

もし、指数や添字の範囲を指定したい場合^{*5}は、その範囲を「{」と「}」(ブレース) とで囲みます。

ブレースによる上付き、下付き文字の範囲の指定 —

$x^{\{10\}}$ とすると、 x^{10} となって嬉しい。これを利用すれば、 $\displaystyle x^{\{x^x\}}$ などとして、 x^{x^x} のような事もできますし、 x_{i-1} と $x_{\{i-1\}}$ とすることにより、 $x_i - 1$ と x_{i-1} を書き分ける事が可能になります。

^{*4} 数式の世界では変数を 1 文字で表す習慣があります。従って、英字が並んでいる場合 (例えば xyz の場合) は、複数の変数の積を表現している (すなわち $x \times y \times z$ の「 \times 」を省略した記法) と解釈される事に注意してください。

^{*5} 逆に言えば、指数や添字の長さが 1 でない場合は、範囲指定が必要なわけですが、一々、文字数を考えるのは面倒ですから、「かならず、範囲指定をする」と「決めておく」と良いでしょう。

この「{」と「}」を利用した、範囲の指定は、色々な所で利用するので、憶えておきましょう。

3.4 関数名

三角関数名などをそのまま書くと、変数の積と区別が付ません。そこで、よく利用される関数に関しては、予め、その関数に対応したマクロが定義されているので、それを利用します。

関数名

`$\sin(\frac{\pi}{4})=\frac{\sqrt{2}}{2}$` と書くと、 $\sin(\frac{\pi}{4}) = \frac{\sqrt{2}}{2}$ となって、関数のようにみえません。

このような場合は、マクロ `\sin` を利用して、`$\sin{(\frac{\pi}{4})}=\frac{\sqrt{2}}{2}$` とすれば、 $\sin(\frac{\pi}{4}) = \frac{\sqrt{2}}{2}$ となって、関数のようにみえます。

4 数学科の科目の数式

数学科一年で学ぶ数学科の科目の数式を $\text{T}_{\text{E}}\text{X}$ で表現してみましょう*⁶。

4.1 微分積分学

微分積分学では、極限や積分などを利用します。

数列の極限は、

極限值 ($\text{T}_{\text{E}}\text{X}$)

```
\[
\lim_{n\rightarrow\infty} \frac{1}{n} = 0
\]
```

とすると、

極限值 (タイプセット後)

$$\lim_{n \rightarrow \infty} \frac{1}{n} = 0$$

となります。

件の $\epsilon - N$ 論法ですが、

*⁶ 担当の先生にお願いすれば、配布されるプリントの $\text{T}_{\text{E}}\text{X}$ のファイルが入手できると思います。是非、お願いしてみよう。

$\epsilon - N$ 論法 (TeX)

```
\[
\lim_{n \rightarrow \infty} a_n = \alpha
\,, \Leftrightarrow \,
\forall \epsilon > 0 \,,
\quad \exists N_0 \,, \text{ s.t. } \,
[
\quad \forall n \geq N_0 \Rightarrow |a_n - \alpha| < \epsilon
]
\]
```

として、

$\epsilon - N$ 論法 (タイプセット後)

$$\lim_{n \rightarrow \infty} a_n = \alpha \Leftrightarrow \forall \epsilon > 0 \exists N_0 \text{ s.t. } [\forall n \geq N_0 \Rightarrow |a_n - \alpha| < \epsilon]$$

といった感じ。

微分は、

微分 (TeX)

```
\[
\frac{d^2 f}{dx^2} = f''(x) = f^{(2)}(x)
\]
```

とすれば、

微分 (タイプセット後)

$$\frac{d^2 f}{dx^2} = f''(x) = f^{(2)}(x)$$

となります。

積分は、

積分 (TeX)

```
\[
\int x \,, \quad dx = x^2 + C \,,
\quad \text{\texttt{\textbackslash mbox{(\$C\$ は積分定数)}}} \,,
\int_0^1 \frac{1}{\sqrt{1-x^2}} = \frac{\pi}{4}
\]
```

とすれば、

積分 (タイプセット後)

$$\int x dx = x^2 + C \text{ (} C \text{ は積分定数)}, \int_0^1 \frac{1}{\sqrt{1-x^2}} = \frac{\pi}{4}$$

となります。 \backslash は、数式モード内で利用し、少し空白を入れるマクロです。数式モードでは、ほとんどの場合空白は無視されるので、明示的に空白を入れたい場合などに重宝します。

4.2 代数幾何

ベクトルや行列は、数が並んでいるものを扱います。このような場合は `\begin{array} ~ \end{array}` を利用します。縦の長さは、行の長さになります。また、横の長さは、`\begin{array}` の後にその個数 (ベクトルの場合は一個なので `{c}` とし、3 列の場合は、その個数を表すために `{ccc}` と、「`c`」の個数で、列の個数を指定します) を表現します。

例えば、ベクトルや行列は次のように表現します。

ベクトルや行列 (T_EX)

```
\[
\overrightarrow{a} =
\left(
\begin{array}{c}
1 \\
2 \\
3
\end{array}
\right)
, \backslash,
A =
\left(
\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}
\right)
\]
```

とすれば、

ベクトルや行列 (タイプセット後)

$$\vec{a} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

となります。