

TEX で色々な数式を記述する

栗野 俊一 *

Ver. 0.05 (2019/10/29 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様になります。

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、
それはとってま{\em 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、{\em 内容の作成に専念}する」
事ができます。

これは、TEX で変換すると次の様に整形され、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、段落を分けて*1くれます。

* 日本大学理工学部数学科 准教授 (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、 $\text{T}_{\text{E}}\text{X}$ 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、 $\text{T}_{\text{E}}\text{X}$ 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。 $\text{T}_{\text{E}}\text{X}$ 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、 $\text{T}_{\text{E}}\text{X}$ 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「 $\text{T}_{\text{E}}\text{X}$ 」と表示される）を記入します。

これらは、 $\text{T}_{\text{E}}\text{X}$ 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。 $\text{T}_{\text{E}}\text{X}$ で学ぶ事の一つは、どの様なマクロがあり、また、それがどの様な影響を持っているかを知る事です。

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、この文章内では、マクロ表現として「\ \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

$\text{T}_{\text{E}}\text{X}$ 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させるかを判断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、数式の両側を「 $\$$ 」で挟みます。例えば、次の様にします。

文中に式を入れる例 ($\text{T}_{\text{E}}\text{X}$)

文中に数式 $y=\frac{1}{x}$ を入れた場合。

*2 これを何故、「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、次の様に表示されます。

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数式を、「`\[`」と「`\]`」で挟みます。

段落としての数式 (TEX)

段落として数式

`\[`

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

`\]`

を入れる場合。

これは、次の様に表示されます。

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。

この様に、`$` と `$` や、`\[` と `\]` に挟まれた部分は、「数式モードになって」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、色々并表示形式が異なります。上記の例の様に分数や、総和記号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。

次の様に段落として、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、次の様に表示されます。

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。
次の様に段落として、記述すれば、綺麗なのですが。

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

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

この様な場合は、`\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、数式は綺麗でも、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 $\displaystyle \frac{1}{x}$ の様にできます。

これは、次の様に表示されます。

`displaystyle` の利用 (タイプセット後)

文中の数式でも $\frac{1}{x}$ の様にできます。

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 の様に「^」を利用し、添字は x_i の様に「_」を利用します。

上付、下付文字 (タイプセット後) —————

指数は、 x^2 の様に「^」を利用し、添字は x_i の様に「_」を利用します。

3.3 範囲指定

上記の「^」や「_」は、直後にある「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、次の様なものです。

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。

^{*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。

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、 $\displaystyle x^{x^{\{x^x\}}}$ などとして、 $x^{x^{x^x}}$ の様な事もできますし、 $\$x_{i-1}\$$ 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。

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

三角関数や指数関数等の初等関数の名前を、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、関数の様にも見えません。

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、関数の様にも見えます。

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)

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

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)

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

とすれば、

積分 (タイプセット後)

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

となります。\\, 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 代数学幾何学 (線形代数)

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様に表現します。

ベクトルや行列 (T_EX)

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

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。

5 Mathematica の利用

数式の計算を手作業でしたくない場合や、そもそも TeX で、その式をどうやって表現してよいか解らない場合^{*7} に、それを *Mathematica* に聞いてしまう という手もあります。

Mathematica には、TeXForm という関数があり、「Mathematica の式」を TeXForm に渡すと、「TeX の式の表現」が表示されます。

後は、これを TeX ファイルの中に Copy & Past するだけです。

6 箇条書き

文章を作成するときには、箇条書は重要な表現の一つです。TeX における箇条書表現には、次の 3 種類があります。

列挙: 記号付き (itemize) 項目を列挙する場合に利用されます。 `\begin{itemize}` で始め、項目の前には `\item` を先行します。最後は、 `\end{itemize}` で閉めます。

列挙 (TeX)

```
\begin{itemize}
\item 林檎
\item 梨
\item 蜜柑
\end{itemize}
```

とすると、

列挙 ((タイプセット後)

- 林檎
- 梨
- 蜜柑

となります。

整列: 番号付き (enumerate) 項目を整列する場合に利用されます。 `\begin{enumerate}` で始め、項目の前には `\item` を先行します。最後は、 `\end{enumerate}` で閉めます。

^{*7} その場合は、当然、「ググる」のが正しいが、その他にも.. という話。

列挙 (TeX)

```
\begin{enumerate}
\item 林檎
\item 梨
\item 蜜柑
\end{enumerate}
```

とすると、

列挙 ((タイプセット後)

1. 林檎
2. 梨
3. 蜜柑

となります。

分類: 項名付き (description) 項目を分類する場合に利用されます。 `\begin{description}` で始め、項目の前には `\item` と、その後ろにオプション引数の「[項目名]」を先行します。最後は、 `\end{description}` で閉めます。

列挙 (TeX)

```
\begin{description}
\item[アダムとイブが食べた] 林檎
\item[千葉県の名産] 梨
\item[冬の次期にはお世話になります] 蜜柑
\end{description}
```

とすると、

列挙 ((タイプセット後)

アダムとイブが食べた 林檎
千葉県の名産 梨
冬の次期にはお世話になります 蜜柑

となります。