

「Short Math Guide for L^AT_EX」を読んで

はじめに

この文書は、「Short Math Guide for L^AT_EX, version 1.09 (2002-03-22) ^{*1}」を自分に合わせて要約したものである。重要と思われること、知らなかったことを中心に抜き出した。

1 Introduction

- $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX 利用する場合、ソースは次のように書く。

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

- 記号全般については、10 節に書いてある。
- ここに載っていない記号が必要な場合は、次を参照する。Comprehensive L^AT_EX Symbols List (Pakin):

<http://www.ctan.org/tex-archive/info/symbols/comprehensive/>

2 Inline math formulas and displayed equations

2.1 The fundamentals

- 数式モードとの切り替えで使うコマンド、環境は次の通り。
inline formulas: $\$ \dots \$$ or $\backslash(\dots \backslash)$.
displayed equations (unnumbered): $\backslash[\dots \backslash]$ or $\backslashbegin{equation*} \dots \backslashend{equation*}$.
displayed equations (automatically numbered) $\backslashbegin{equation} \dots \backslashend{equation}$
- 注意点: plain T_EX で使われる $\mathcal{E}\mathcal{E} \dots \mathcal{E}\mathcal{E}$ の利用は推奨されない。
- 数式群や複数行からなる数式を記述するための環境は次の通り。
equation と split の組み合わせ split 環境は, equation, align, gather 環境などの内部においてのみ有効。

^{*1} <http://www.ams.org/tex/amslatex.html> にある。

```

\begin{equation}\label{xx}
\begin{split}
a &= b+c-d \\
& \quad + e-f \\
& \quad = g+h \\
& \quad = i
\end{split}
\end{equation}

```

multiline 数行にわたる長い数式のときに利用する.

```

\begin{multiline}
a+b+c+d+e+f \\
+i+j+k+l+m+n
\end{multiline}

```

$$a + b + c + d + e + f + i + j + k + l + m + n \quad (2.2)$$

gather 数式をまとめてセンタリング.

```

\begin{gather}
a_1=b_1+c_1 \\
a_2=b_2+c_2-d_2+e_2
\end{gather}

```

$$a_1 = b_1 + c_1 \quad (2.3)$$

$$a_2 = b_2 + c_2 - d_2 + e_2 \quad (2.4)$$

align 等号部分などをそろえるとき.

```

\begin{align}
a_1&=b_1+c_1 \\
a_2&=b_2+c_2-d_2+e_2
\end{align}

```

$$a_1 = b_1 + c_1 \quad (2.5)$$

$$a_2 = b_2 + c_2 - d_2 + e_2 \quad (2.6)$$

```

\begin{align}
a_{11} &= b_{11} & a_{12} &= b_{12} \\
a_{12} &= b_{12} \\
a_{21} &= b_{21} & a_{22} &= b_{22} + c_{22} \\
a_{22} &= b_{22} + c_{22}
\end{align}

```

$$a_{11} = b_{11} \quad a_{12} = b_{12} \quad (2.7)$$

$$a_{21} = b_{21} \quad a_{22} = b_{22} + c_{22} \quad (2.8)$$

flalign* 複数の&がある場合, 左右に広げて表示する.

```

\begin{flalign*}
a_{11}&=b_{11} & & & a_{12} &= b_{12} \\
a_{12}&=b_{12} \\
a_{21}&=b_{21} & & & a_{22} &= b_{22} + c_{22} \\
a_{22}&=b_{22}+c_{22}
\end{flalign*}

```

$$a_{11} = b_{11} \quad a_{12} = b_{12}$$

$$a_{21} = b_{21} \quad a_{22} = b_{22} + c_{22}$$

- split は, 複数行にわたる数式を一行の数式のように取り扱う (グループ化を行なう) ものと考えればいいだろう.
- eqnarray 環境と eqnarray* 環境の利用は推奨されない.

2.2 Automatic numbering and cross-referencing

- 相互参照のためのラベルの張り方: `\begin{equation}\label{reio} ... \end{equation}`
- 参照には`\eqref`を使う:

... using equations `\eqref{ax1}` and `\eqref{bz2}`, we can derive ...

こうすると、次のようになる。

using equations (3.2) and (3.5), we can derive

つまり、`\eqref{ax1}`と`(\ref{ax1})`は同等。

- 数式番号を(節番号. 式番号)とするには、プリアンブルで`\numberwithin`を使う。例えば、次のように記述する。`\numberwithin{equation}{section}`.
- その他については、Lamport 先生の本を見る [Lamport, 6.3, C.8.4].
- `subequations` 環境は次のように使える。現在の数式番号が 2.1 のとき、

```
\begin{equation}\label{first}
a=b+c \end{equation}
some intervening text
\begin{subequations}\label{grp}
\begin{align}
a&=b+c\label{second}\\
d&=e+f+g\label{third}\\
h&=i+j\label{fourth}
\end{align}
\end{subequations}
```

と書くと、次のような出力を得る。

$$a = b + c \tag{2.9}$$

some intervening text

$$a = b + c \tag{2.10a}$$

$$d = e + f + g \tag{2.10b}$$

$$h = i + j \tag{2.10c}$$

`\eqref{grp}`によって (2.10) を、`\eqref{second}`によって (2.10a) を得る。

3 Math symbols and math fonts

3.1 Classes of math symbols

数式における記号は、異なるクラスに分けられている。クラスごとに(文字)間隔・位置が決められている。

クラスの番号	Mnemonic	種類 (品詞)	例
0	Ord	simple/ordinary (“noun”)	$A 0 \Phi \infty$
1	Op	prefix operator	$\Sigma \Pi \int$
2	Bin	binary operator (conjunction)	$+ \cup \wedge$
3	Rel	relation/comparison (verb)	$= < \subset$
4	Open	left/opening delimiter	$([\{ $
5	Close	right/closing delimiter	$)] \} $
6	Pun	postfix/punctuation	$. , ; !$

- クラス Bin の記号 (特にマイナス記号 $-$) は、もし自分の左側のオペランドがなければ自動的にクラス 0 となる。例: -2 と $5-2$ 。
- 文字間隔は、一般的な規則ではなく伝統的な規則にしたがっている場合がある。例えば、 $/$ はクラス 2 であるが、 $k/2$ では、スラッシュの前後のスペースがない。

3.2 Some symbols intentionally omitted here

\LaTeX で用意されている記号のうち、`amssymb` パッケージを使うことによって別名で利用できるものについての解説は省略している。例えば、 \square は、 \LaTeX では `\Box`、`amssymb` パッケージをロードしていると `\square` で利用できる。完全な記号リストについては、The Comprehensive \LaTeX Symbols List (Pakin):

<http://www.ctan.org/tex-archive/info/symbols/comprehensive/>

を見ましょう。

3.3 Latin letters and Arabic numerals

- ラテン語文字はクラス 0。数式モードでは、*italic* になる。
- ドットがない i や j について: `\imath` で i 、`\hat{\jmath}` で \hat{j} など。
- アラビア数字はクラス 0。デフォルトで `upright/roman` となる。

3.4 Greek letters

- ラテン語文字と同様にギリシャ文字はクラス 0。
- 数式モードの中で、ギリシャ文字の小文字は *italic*、大文字は `upright/roman`。

3.5 Other alphabetic symbols

その他のクラス 0 のもの。例えば、 \aleph 、 \beth など。

3.6 Miscellaneous simple symbols

- その他のクラス 0 のもの。

- \square や $\#$ の利用において普通にある間違いは、それらを演算子 (binary operators) や関係演算子 (binary relation) として使ってしまうことである。しかし、これらは、クラス 0 であるから、単純にそのような利用をしてしまうと、間違った文字間隔になる。例: $a\square b$.

3.7 Binary operator symbols

$*$, \cdot , \wedge など.

3.8 Relation symbols: $< = > \prec \sim$ and variants

3.9 Relation symbols: arrows

4 節も見よ。 \circ , \Leftarrow , \nearrow など.

3.10 Relation symbols: miscellaneous

\exists , \nexists , \sim など.

3.11 Cumulative (variable-size) operators

\int , \odot , \oplus など.

3.12 Punctuation

- \cdot , $;$, \cdots など.
- 「:」 そのものはクラス 3(relation) の文字間隔を作る。 `\colon` コマンドとの違いは、次を見れば明らか。
 - `\colon` を使った場合: $f: A \rightarrow B$
 - `:` を使った場合: $f : A \rightarrow B$
- `\cdots` や `\ldots` がしばしば使われるが、ほとんどの場合は、`\dotsb`, `\dotsc`, `\dotsi`, `\dotsm`, `\dotso` を推奨する。4.6 を見よ.

3.13 Pairing delimiters (extensible)

詳しくは、6 節を見よ.

3.14 Nonpairing extensible symbols

`\vert`, `|`, `\Vert`, `\lvert` を paired delimiters として使うことを推奨しない (6.2 を見よ).

3.15 Extensible vertical arrows

3.16 Accents

3.17 Named operators

- named operator を定義するには、`\DeclareMathOperator` コマンドを利用する。例えば、次のように
プリアンブルに書いた場合、

```
\DeclareMathOperator{\rank}{rank}
\DeclareMathOperator{\esssup}{ess\,sup}
```

`\rank(x)` で $\text{rank}(x)$, `\esssup(y,z)` で $\text{ess sup}(y,z)$ を得る。ディスプレイ方式対応にするには、`\DeclareMathOperator*` コマンドを利用する。

- 直接指定する方法もある。例えば、`\operatorname{rank}(x)` とすると、 $\text{rank}(x)$ を得る。

3.18 Math font switches

- ラテン語の大文字だけがフォント・スイッチのすべてに正確に応じる。
- 数学記号は、ほぼすべてフォント・スイッチに影響されない。
- ラテン語の小文字、ギリシアの大文字および数字は、いくつかのフォント・スイッチに適切に応じるが、奇妙な結果になるところもある。
- Lucida New Math ような代替数学フォント・セットの使用すると、多少改善するかもしれない。
- `\mathbf` が適用できない記号に対しては、`\boldsymbol` または `\pmb` コマンドが使える。

$$A_\infty + \pi A_0 \sim \mathbf{A}_\infty + \pi \mathbf{A}_0 \sim \mathbf{A}_\infty + \pi \mathbf{A}_0 \quad (3.1)$$

`A_\infty + \pi A_0`

`\sim \mathbf{A}_\infty + \pi \mathbf{A}_0`

`\sim \mathbf{A}_\infty + \pi \mathbf{A}_0`

`\sim \pmb{A}_\infty + \pi \pmb{A}_0`

- `\boldsymbol` コマンドは、`bm` パッケージを使うことによって (よりよいものが?) 得られる。`bm` パッケージは、`amsmath` パッケージが提供するものより新しく強力なバージョンを供給する。(`bm.sty` の最後の方に `\let\boldsymbol\bm` と書いてある。したがって、`bm` パッケージを使った場合、`\bm` コマンドが置き換えられることがわかる。)
- 一度に一つの記号に `\boldsymbol` コマンドを適用することが原則。

3.18.1 Calligraphic letters (`cmsy`; no lowercase)

`\mathcal{M}`: \mathcal{M}

3.18.2 Blackboard Bold letters (msbm; no lowercase)

`\mathbb{R}`: \mathbb{R}

3.18.3 Fraktur letters (eufm)

`\mathfrak{S}`: \mathfrak{S}

4 Notations

4.1 Top and bottom embellishments

見た目にはアクセントに似ているが、延びるもの. \tilde{x} , \widetilde{xx} , \widetilde{xxx} , \widetilde{xxxx} , \widetilde{xxxxx} , \widetilde{xxxxxx} , $\widetilde{xxxxxxx}$.

4.2 Extensible arrows

`\xleftarrow` と `\xrightarrow` は、上下の文字列によって自動的に延びる矢印. 例えば、

`A\xleftarrow{n+\mu-1}B\xrightarrow[T]{n\pm i-1}C`

と書けば、次を得る.

$$A \xleftarrow{n+\mu-1} B \xrightarrow[T]{n\pm i-1} C$$

4.3 Affixing symbols to other symbols

`\overset` と `\underset` を使うと、3.16 節の標準的なアクセントの他のものを記号の上下に置くことができる. 例えば、`\overset{*}{X}` と書けば、 X^* を得る. 8.4 節の `\sideset` も参照せよ.

4.4 Matrices

- `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix`, `Vmatrix` 環境は、いずれも行列の環境であるが、それぞれ、区切り記号が、`()`, `[]`, `{ }`, `||`, `|||` である.
- `matrix` 環境は、区切り記号なしの行列.
- 小さい行列を得るには、`smallmatrix` 環境を使う. 例えば、 $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ をには、次のようにする.

```
\bigl( \begin{smallmatrix}
a&b \\
c&d \end{smallmatrix} \bigr)
```

- 点々の行を作るには、`\hdotsfor{3}` などとする. 例えば、 $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & \dots & \dots & \dots \end{bmatrix}$.
- 関数の定義などで使われる `cases` 環境: $P_{r-j} = \begin{cases} 0 & \text{if } r-j \text{ is odd,} \\ r! (-1)^{(r-j)/2} & \text{if } r-j \text{ is even.} \end{cases}$

```
P_{r-j}=\begin{cases}
```

```

0& \text{if $r-j$ is odd},\
r!\,(-1)^{(r-j)/2}& \text{if $r-j$ is even}.
\end{cases}

```

- plain TeX の `\matrix` コマンド, `\pmatrix`, `\cases` コマンドは, `amsmath` パッケージがロードされる
とき無効になる.

4.5 Math spacing commands

- `amsmath` パッケージを使うと, スペース調整コマンドが, 数式モードの中でも外でも利用できる. 生の
LaTeX では, `\;` (`\thickspace`) などは, 数式モード内でないと利用できない.
- より繊細にスペース調整をするには, `\mspace` コマンドと「math units」を利用する. `mu` は $1/18em$ に
等しい. よって, `\mspace{-9.0mu}` で, `\quad` の半分の負のスペースを得ることができる.
- 各文字列 (など) と同じスペースを空ける LaTeX のコマンド:

例	結果
<code></code>	三つの X と同じ幅, 同じ高さのスペース
<code>\hphantom{XXX}</code>	三つの X と同じ幅で高さが 0 のスペース
<code>\vphantom{X}</code>	X と同じ高さで幅が 0 のスペース

4.6 Dots

一般的な約束ごとはないようだが, `\ldots` や `\cdots` を使うのではなくて, 次のコマンドを使った方が良さ
そう.

- `\dotsc`: コンマ (commas) と共に用いる点々
- `\dotsb`: binary operators/relations と共に用いる点々
- `\dotsm`: 掛け算を省略するときの点々
- `\dotsi`: integrals と共に用いる点々
- `\dotso`: その他の点々 (上記に当てはまらないもの)

4.7 Nonbreaking dashes

- `\nobreakdash` コマンドによってハイフン, ダッシュのところでの改行を抑圧することができる. 例え
ば, 「pages 1–9」 (`pages 1\nobreakdash--9`) と書いた場合, ダッシュと 9 の間では決して改行は起
らない.
- このコマンドは, 「`p-adic`」といった組み合わせにおいて, 望ましくないハイフネーションを起こさ
せないようにするためにも利用できる.
- 短縮形をつくっておくことは賢明である. 例えば,
 - `\newcommand{\p}{p\nobreakdash}`として `\p-adic`
 - `\newcommand{\Ndash}{\nobreakdash\textendash}`として `pages 1\Ndash 9`
 - `\newcommand{\n}[1]{\n$\nobreakdash-\hspace{0pt}}`として `\n dimensional`. この例で

は、ハイフンのあとでの改行を禁止するが、ハイフンの後の語では普通に許されるようにする方法が示されている。(幅 0 のスペースをハイフンの後に付け加えるだけで十分である.)

4.8 Roots

`\sqrt[3]{2}`: $\sqrt[3]{2}$ など.

4.9 Boxed formulas

`\boxed` コマンドを使うと数式モード内で、引数のものを箱にいれることができる。`\fbox` コマンドに似たようなものと考えれば良い。数式番号を含む箱を必要とする場合は、`amsmath` パッケージの FAQ を見よ。

5 Fractions and related constructions

5.1 The `\frac`, `\dfrac` and `\tfrac` commands

`\dfrac`: 常に `display-style` の分数, `\tfrac`: 常に `text-style` の分数.

$$\frac{1}{k} \log_2 c(f) \quad \frac{1}{k} \log_2 c(f) \tag{5.1}$$

$$\Re z = \frac{n\pi \frac{\theta + \psi}{2}}{\left(\frac{\theta + \psi}{2}\right)^2 + \left(\frac{1}{2} \log \left| \frac{B}{A} \right| \right)^2}. \tag{5.2}$$

5.2 The `\binom`, `\dbinom`, and `\tbinom` commands

2 項係数についての三つのコマンド.

$$2^k - \binom{k}{1} 2^{k-1} + \binom{k}{2} 2^{k-2} \tag{5.3}$$

5.3 The `\genfrac` command

- `\frac`, `\binom` とその仲間は、一般化された `\genfrac` コマンドの一部と見なされる。そして、このコマンドは、六つの引数をとる。
- 引数のうち最後の二つは、`\frac` の分子 (numerator) と分母 (denominator) である。
- 最初の二つは、省略が可能な区切り文字を指定する。`\binom` で使われている。
- 三番目は、分数の線の太さ。`\binom` では、0pt に設定している。引数を与えないと線の太さはのデフォルトは、「normal」。
- 4 番目は、スタイルの指定。0 から 3 の整数で指定し、それぞれ、`\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle` に対応する。

- 例

```
\newcommand{\frac}[2]{\genfrac{}{}{#1}{#2}}
\newcommand{\tfrac}[2]{\genfrac{}{}{1}{#1}{#2}}
\newcommand{\binom}[2]{\genfrac{}{}{0pt}{#1}{#2}}
```

- 技術的な理由から、`\over`、`\atop`、`\above` といった分数コマンドの利用は勧められない (例えば `amsmath.faq` を見よ).

5.4 Continued fractions

連分数 (continued fraction) には、`\cfrac` を使うとよい.

6 Delimiters

6.1 Delimiter sizes

`\left` と `\right` によってデリミタの大きさが自動的に調整されるが、機械的にやってしまうことなので、場合によっては、`\bigl` などを使って、自分で調節する必要がある.

- `\left` と `\right` を使い必要以上に大きくなってしまいう例と調整した例:

$$\left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p} \quad \text{と} \quad \left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p}$$

(後者は、`\biggl[\sum_i a_i \Bigl| \lvert \sum_j x_{ij} \rvert \Biggr]^p \biggr]^{1/p}`)

- `\left` と `\right` では、全部同じ大きさになってしまう例と入れ子を見やすくするためにデリミタの一部を少し大きくしている例:

$$((a_1 b_1) - (a_2 b_2))((a_2 b_1) + (a_1 b_2)) \quad \text{versus} \quad ((a_1 b_1) - (a_2 b_2))((a_2 b_1) + (a_1 b_2))$$

```
\left((a_1 b_1) - (a_2 b_2)\right)
```

```
\left((a_2 b_1) + (a_1 b_2)\right)
```

```
\quad\text{versus}\quad
```

```
\bigl((a_1 b_1) - (a_2 b_2)\bigr)
```

```
\bigl((a_2 b_1) + (a_1 b_2)\bigr)
```

- `\left` と `\right` では、わずかに大きすぎる例と調節した例:

$\left| \frac{b'}{d'} \right|$ と $\left| \frac{b'}{d'} \right|$ (`\bigl\lvert \frac{b'}{d'} \bigr\rvert`).

6.2 Vertical bar notations

- `|` を組になったデリミタに使用することは勧められない. 間違ったスペーシングによって記号の方向について曖昧さを生じてしまうことが稀にある. これを防ぐには、`\lvert`(left vert bar) や `\rvert`(right vert bar) を用いればよい. この違いは、 $|-k|$ (`| -k|`) と $|-k|$ (`\lvert -k\rvert`) を比較すれば明らか.

- ノルムなどに使用する二本の縦棒には、`\lVert` と `\rVert` を使用する。
- プリアンブルで次のようなコマンドを定義しておくことを推奨する。

```
\providecommand{\abs}[1]{\lvert#1\rvert}
\providecommand{\norm}[1]{\lVert#1\rVert}
```

こうしておけば、`\abs{z}` で $|z|$ を、`\norm{v}` で $\|v\|$ を得る。

7 The `\text` command

`\text` コマンドは、`\mbox` コマンドと似ている。しかし、`\text` コマンドが、subscript で使用されたとき、自動的に subscript サイズのテキストになるのに対し、`\mbox` ではそうならない。

7.1 `\mod` and its relatives

特別なスペーシング習慣がある `\mod`、`\bmod`、`\pmod`、`\pod` コマンドについて。

8 Integrals and sums

8.1 Altering the placement of limits

- 積分記号や和の記号などの範囲の位置を変更するために、三つのコマンドがある。
- `\limits` コマンドを使った例: $\int_{|x-x_z(t)|<X_0} \dots$
- `\nolimits` コマンドを使った例: $\sum_{n=1} \dots$
- `\displaylimits` は、新しく \sum のような記号を定義した際に利用する。
- [AMUG] にある `intllimits` と `nosumlimits` オプションの解説についても参照のこと。

8.2 Multiple integral signs

`\iint` コマンドなど。

8.3 Multiline subscripts and superscripts

複数行にわたる添字には `\substack` コマンドを使う。例えば、

```
\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}}
P(i, j)
```

で、 $\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i, j)$ となる。

8.4 The `\sideset` command

- `\sideset` は、 \sum の肩に添字をつけたいときなどに利用する。このコマンドは、大きいオペレータ文字用に設計されている。普通の文字と共に用いた場合の結果はあてにならない。
- `\sideset{}{'}\sum_{n<k,\;\text{\$n\$ odd}} nE_n` で $\sum'_{n<k, n \text{ odd}} nE_n$ を得る。
- `\sideset` は、大きなオペレータの両側に文字をおくことができる。例えば、 \prod^* を得るには、`\sideset{_*^*}{_*^*}\prod` とする。

9 Changing the size of elements in a formula

\LaTeX では、フォントの大きさを変更する仕組みが、数式内と数式の外では完全に異なる。よって数式の中で、`\large` などは、使用できない (意味がない)。例えば、 $\#$ を \sum 記号とと同じように利用したい場合、それを実現する一番よい方法は、 \LaTeX の標準の `\DeclareMathSymbol` コマンドを用いて、その記号を “mathop” として定義することである ([LFG] を見よ)。

$$\frac{\sum_{n>0} z^n}{\prod_{1\leq k\leq n} (1-q^k)}$$

を考えてみる。 `frac` を `dfrac` に変更しても、この場合は何も変わらない。

$$\frac{\sum_{n>0} z^n}{\prod_{1\leq k\leq n} (1-q^k)}$$

もし、和と積の記号を大きくしたいのならば、`\displaystyle` コマンドを次のように使う必要がある。

$$\frac{\sum_{n>0} z^n}{\prod_{1\leq k\leq n} (1-q^k)}$$

そして、記号は大きいままで、範囲を脇に表示したい場合は、`\nolimits` コマンドを共に用いる。

$$\frac{\sum_{n>0} z^n}{\prod_{1\leq k\leq n} (1-q^k)}$$

これに似たコマンド、つまり、数式において強制的に記号の大きさ (とスペーシング) を変更するコマンドは、この他に、`\textstyle`、`\scriptstyle` と `\scriptscriptstyle` がある。

これらのコマンドを使うときに、コマンドの影響範囲を決める中括弧をどこに置くかに注意する必要がある。

正しい使い方: `{\displaystyle ...}` 間違った使い方: `\displaystyle{...}`

10 Other packages of interest

多くの他の \LaTeX パッケージが、CTAN (the Comprehensive \TeX Archive Network) から入手可能である。いくつかを推薦しておこう。

accents 任意の記号を利用するアクセントなど.
amsthm 定理や証明の設定.
bm Bold math パッケージ, `\boldsymbol` をもっと一般的にする.
cases 個々の数式番号を失うことなく二つ以上の式に対して中括弧を適用する.
delarray array の複数行にまたがるデリミタ.
kuvio 可換図式と他の図式.
xypic 可換図式と他の図式.
rsfs Ralph Smith's Formal Script, フォント設定.

パッケージ名を知りたいならば, \TeX カタログ

<http://www.tex.ac.uk/tex-archive/help/Catalogue/catalogue.html>,

を見るとよい.

11 Other documentation of interest

参考文献

- [AMUG] American Mathematical Society: User's Guide for the amsmath package, `amslatex.tex`, 1999.
- [CLSL] Pakin, Scott: The Comprehensive \LaTeX Symbols List, <http://www.ctan.org/tex-archive/info/symbols/comprehensive/>, July 2001.
- [Lamport] Lamport, Leslie: \LaTeX : a document preparation system, 2nd edition, Addison-Wesley, 1994.
- [LC] Goossens, Michel; Mittelbach, Frank; Samarin, Alexander: The \LaTeX Companion, Addison-Wesley, 1994.
- [LFG] \LaTeX 3 Project Team: \LaTeX 2 ϵ font selection, `fntguide.tex`, 1994.
- [LGC] Goossens, Michel; Rahtz, Sebastian; Mittelbach, Frank: The \LaTeX Graphics Companion, Addison Wesley Longman, 1997.
- [LGG] Carlisle, D. P.: Packages in the 'graphics' bundle, `grfguide.tex`, 1995.
- [LUG] \LaTeX 3 Project Team: \LaTeX 2 ϵ for authors, `usrguide.tex`, 1994.