都数共通パッケージ tosuustd2019 の紹介

樋川 達郎（都数 2019 年度執行部役員，編集長）

2019 年 3 月 9 日

1　概要

発表のレジュメなどの LaTeX 文書の作成を容易にするため，また『数学のなかま』の編集作業を少しでも楽にするために，都数共通のパッケージ tosuustd2019 を作ってみました[*1]．

tosuustd2019 は，pLaTeX 2_ε + dvipdfmx で使うことができます．それ以外の形式，たとえば upLaTeX，XeLaTeX，LuaLaTeX や，dvipdfmx 以外のドライバには対応していません．tosuustd2019 を使うには，tosuustd2019.sty を tex ファイルと同じディレクトリに配置し，tex ファイルには

```
\documentclass[dvipdfmx]{jsarticle}
\usepackage{tosuustd2019}
% 必要なら追加でパッケージを読み込んだり，コマンドを定義したりする

\title{...}
\author{...}
\date{...}

\begin{document}
  % 本文
\end{document}
```

のように書けば OK です．

[*1] 『数学のなかま』の編集方針はその年度ごとの編集部によって違ってくると思うので，2019 と明示してあります．来年度に「別のコマンド群を使いたい」となれば，tosuustd2020 などが作られることもあるかもしれません（もちろん，2019 のものをそのまま使ってもらっても構いません）．

tosuustd2019 の内部では，大きく分けて，次の 2 つの処理を行っています．

(1) 使用頻度が高い（と思われる）パッケージの読み込み（3 節）
(2) 数学文書の作成にあたって，あると便利な（と思われる）マクロの定義（4 節）

tosuustd2019 を読み込めば，プリアンブルでたくさんのパッケージを \usepackage する必要はなくなり，文書の作成・編集作業の簡略化に繋がります．また，tosuustd2019 で定義されるマクロを使ってもらうことで，「間違った」入力を減らすことも期待できます．たとえば，数式モード内で写像の記号を「f : X \to Y」と書くのはよくある間違いですが（: ではなく \colon とすべきです），tosuustd2019 で定義されるマクロ map を使って \map{f}{X}{Y} と書けば，自動的に正しい出力が得られます．

2 オプション

tosuustd2019 に指定できるオプションについて説明します．オプションを指定するには，読み込みの際に \usepackage[〈オプションの指定〉]{tosuustd2019} のように書きます．

■slantedGreek オプション　TEX の既定では，慣習に従って，数式中のギリシア大文字はイタリック体ではなく立体で出力されます．これをイタリック体で出力するようにするのが，slantedGreek オプションです．たとえば，数式モード内で \Gamma と入力した場合，既定では Γ と出力されますが，このオプションを指定した場合は $\mathit{\Gamma}$ と出力されます．

なお，tosuustd2019 では LATEX にないギリシア文字のコマンド（\Alpha など．ラテン文字と同じ形なので省かれている）を新たに定義していますが (4.1 節)，slantedGreek オプションの効果はこれにも及びます．

3 パッケージの読み込み

tosuustd2019 では，次のパッケージを読み込んでいます．

3.1 欧文フォント関連

fontenc　T1 オプションを指定して読み込んでいます．フォントエンコーディングを，TEX の既定の OT1 から T1 に変更します．（OT1 エンコーディングのままだと，アクセント付きアルファベットのカーニングがうまくいかないなどの問題があります．詳しくは，『美文書』 [1, p. 210] や解説記事 [3] を参照してください．）

textcomp　TS1 エンコーディングの文字群を使うためのコマンドを定義します．『美文書』 [1, p. 388–391] に使えるコマンドの一覧表があります．

inputenc　utf8 オプションを指定して読み込んでいます．ASCII 範囲外の欧文文字を，UTF-8 エンコーディングで直接入力できるようになります．詳しくは，『美文書』 [1, p. 213–214] や解説記事 [4] を参照してください．（実は，TEX Live 2018 以降では，既定で UTF-8 入力が有効になっています [5]．したがって，TEX Live 2018 以降ではこの記述は不要ということになりますが，そうでない環境のことも考えて，inputenc パッケージを読み込んでいます．）

lmodern　欧文に Latin Modern（フォント）を使用します．見た目は TEX 既定の Computer Modern と同じですが，様々な改良が施されています．

tgheros　欧文のサンセリフ体に TEX Gyre Heros を使用します．Helvetica を元にしたフォントです．

3.2　数学関連

mathtools　内部で amsmath パッケージを読み込み，さらに若干の修正・拡張をするパッケージです．

amssymb　様々な数学記号を含む AMSFonts を使うためのパッケージです．

bm　数式中でイタリック体の太字を出力するためのコマンドを定義します．

mathrsfs　数式の筆記体フォントである RFSF (Ralph Smith's Formal Script) を，\mathscr で使えるようにします．

mathcomp　textcomp パッケージの記号を数式モードでも出すためのマクロを定義します．特に，\tcdegree で角度の記号 ° を出すことができます．

amsthm　定理環境をカスタマイズするためのマクロや，証明環境を提供してくれるパッケージです．amsthm パッケージ既定の theoremstyle は欧文用のもので和文には馴染まないので，パッケージを読み込んだあとに theoremstyle を再定義しています．また，既定の proof 環境もやはり欧文向けなので，これも再定義しています．定理環境・証明環境については，4.5 節も参照してください．

3.3　和文フォント関連

otf　jis2004 オプションを指定して読み込んでいます．Unicode 番号や CID 番号によって文字にアクセスするコマンドが定義されたり，和文のフォントメトリックが改良されたりします．詳しくは，『美文書』[1, p. 263–269] や解説記事 [3] を参照してください．

3.4　箇条書き

enumitem　箇条書きのカスタマイズを容易にするためのパッケージです．なお，enumitem パッケージは箇条書きのスタイルを欧文用のものに変えてしまうので，パッケージを読み込んだあとに再定義しています．さらに，このパッケージの機能を使って，箇条書きの環境をいくつか定義しています（4.7 節）．

3.5　グラフィック，表組みなど

graphicx　グラフィックを扱うためのパッケージです．たとえば，画像が挿入できるようになります．詳しくは，『美文書』[1, 第 7 章] を参照してください．

xcolor　dvipsnames，table オプションを指定して読み込んでいます．色を扱うためのパッケージです．詳しくは，『美文書』[1, p. 139–142] を参照してください．（table オプションを指定することで，内部で colortbl パッケージが読み込まれ，表にも色を付けることができるようになります．これについては，『美文書』[1, p. 152–153] を参照してください．）

float　図・表の配置に関する機能を強化してくれます．

array　LATEX 標準の表組み機能を改良してくれます．

booktabs　整った横罫線を出力するためのコマンドを定義します．

tabularx　横幅が決まった表を出力するための，tabularx 環境を定義します．

longtable　ページをまたぐ表を出力するための，longtable 環境を定義します．

multirow　表で，縦方向にセルを結合できるようにします．

tikz　強力な図形描画コマンド群です．tikz パッケージを読み込んだあとに，\usetikzlibrary で arrows，calc，intersections，patterns，quotes，shapes，through を読み込んでいます．詳しくは，『美文書』 [1, 付録 D] や TEX Wiki [7] を参照してください．

tikz-cd　可換図式を書くためのパッケージです．

mdframed　枠囲みをするためのパッケージです．

3.6　ハイパーリンク，相互参照

hyperref　ハイパーリンクを使用するためのパッケージです．オプションは，dvipdfmx，bookmarks=true，bookmarksnumbered=true，setpagesize=false を指定しています（『美文書』 [1, p. 179]，TEX Wiki [6] を参考にしました）．また，このパッケージを読み込んだ後（tosuustd2019 を読み込んだ後）に

```
\hypersetup{%
  pdftitle=...,%
  pdfsubject=...,%
  pdfauthor=...,%
  pdfkeywords=...}
```

のように書くことで，PDF ファイルに文書情報を付けることができます．

pxjahyper　hyperref パッケージは PDF ファイルにしおりを付けてくれますが，そのままでは日本語が文字化けしてしまいます．pxjahyper パッケージは，これを修正してくれます．

cleverref　「賢い」相互参照用のコマンド \cref を定義します．たとえば定理を参照するとき，LATEX 標準の \ref コマンドでは「定理~\ref{thm:euler}」のように書く必要がありますが，\cref コマンドでは「\cref{thm:euler}」のように書くだけで済みます（自動で「定理」を補ってくれます）．

4　マクロの定義

tosuustd2019 では，次のマクロを定義しています．

4.1　数式で使う文字

LATEX では，ラテン文字と同じ形をしているギリシア文字は，コマンドとして定義されていません．tosuustd2019 では，これを補っています．

\Alpha	A	\Iota	I	\omicron	o
\Beta	B	\Kappa	K	\Rho	P
\Epsilon	E	\Mu	M	\Tau	T
\Zeta	Z	\Nu	N	\Chi	X
\Eta	H	\Omicron	O		

これらのコマンドの使い道は限られると思いますが，たとえばベータ関数は B(p, q) よりも \Beta(p, q) と書くほうがよいでしょう（こうすることで，slantedGreek オプションの効果が及ぶようになります）．

特殊な書体の数式文字を簡単に出せるようにするために，次のコマンドを定義しています[*2]．

\mbfA, \dots, \mbfZ, \mbfa, \dots, \mbfz → $\mathbf{A}, \dots, \mathbf{Z}, \mathbf{a}, \dots, \mathbf{z}$

\mbfitA, \dots, \mbfitZ, \mbfita, \dots, \mbfitz → $\boldsymbol{A}, \dots, \boldsymbol{Z}, \boldsymbol{a}, \dots, \boldsymbol{z}$

\mcalA, \dots, \mcalZ → $\mathcal{A}, \dots, \mathcal{Z}$

\mscrA, \dots, \mscrZ → $\mathscr{A}, \dots, \mathscr{Z}$

\mfrakA, \dots, \mfrakZ, \mfraka, \dots, \mfrakz → $\mathfrak{A}, \dots, \mathfrak{Z}, \mathfrak{a}, \dots, \mathfrak{z}$

また，特定の集合を表すのによく使われる黒板太字を出すために，次のコマンドを定義しています．

\setN, \setZ, \setQ, \setR, \setC, \setH, \setK → $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}, \mathbb{H}, \mathbb{K}$

4.2 括弧など

数式中で伸縮する括弧を出すためには，\left(...\right)，\left[...\right]，\left\{...\right\} のように書きますが，これらをそれぞれ \parenlr{...}，\bracketlr{...}，\bracelr{...} と書けるようにしました．たとえば，別行立て数式で

```
\parenlr{\sum_{k = 1}^{n} x_k y_k}^2
\leq \parenlr{\sum_{k = 1}^{n} x_k^2} \parenlr{\sum_{k = 1}^{n} y_k^2}
```

と入力すると，

$$\left(\sum_{k=1}^{n} x_k y_k\right)^2 \leq \left(\sum_{k=1}^{n} x_k^2\right) \left(\sum_{k=1}^{n} y_k^2\right)$$

と出力されます．

丸括弧，角括弧，波括弧以外の括弧類を出すためのコマンドとして，次の 5 つを定義しています．

\abracket{x}	$\langle x \rangle$	\floor{x}	$\lfloor x \rfloor$	\abs{x}	$\lvert x \rvert$
		\ceil{x}	$\lceil x \rceil$	\norm{x}	$\lVert x \rVert$

つまり，\langle x \rangle の代わりに \abracket{x} と書くことができます．また，\abracketlr で，伸縮する角括弧を出力できます．\floorlr，\ceillr，\abslr，\normlr も同様です．

\set で集合の内包的記法が，\map で写像の記号が，\restr で写像の制限の記号が出せます．

[*2] これらのコマンド名は，unicode-math（XeLaTeX および LuaLaTeX 用のパッケージ）に合わせました [2].

$$\begin{array}{lcl} \texttt{\textbackslash set\{x\}\{P(x)\}} & \to & \{x \mid P(x)\} \\ \texttt{\textbackslash map\{f\}\{X\}\{Y\}} & \to & f\colon X \to Y \\ \texttt{\textbackslash restr\{f\}\{A\}} & \to & f|_A \end{array}$$

`\setlr`, `\restrlr` はそれぞれ `\set`, `\restr` の「伸縮する」版です.

`\id` で恒等写像の記号 id が出せます.

4.3 log 型関数

次の「log 型関数」を定義しています[*3].

`\Aut`	Aut	`\coker`	coker	`\ImPart`	Im
`\Coim`	Coim	`\colim`	colim	`\Ker`	Ker
`\coim`	coim	`\End`	End	`\lcm`	lcm
`\Cok`	Cok	`\Hom`	Hom	`\rank`	rank
`\cok`	cok	`\Image`	Im	`\RePart`	Re
`\Coker`	Coker	`\image`	im	`\tr`	tr

`\Image` と `\ImPart` は，出力は同じですが，前者は像を表す記号として，後者は虚部を表す記号として使われることを意図しています.

4.4 論理記号

テキスト中で `\IFF{(a)}{(b)}` と書けば，(a) $\Longleftrightarrow$ (b) と出力されます．同様に，`\IMPLIES{(a)}{(b)}` と書けば (a) $\Longrightarrow$ (b)，`\IMPLIEDBY{(a)}{(b)}` と書けば (a) $\Longleftarrow$ (b) と出力されます.

4.5 定理環境，証明環境

定理環境として，次の 12 個を定義しています.

環境名	種類	環境名	種類	環境名	種類
`theorem`	定理	`claim`	主張	`remark`	注意
`proposition`	命題	`fact`	事実	`example`	例
`corollary`	系	`definition`	定義	`problem`	問題
`lemma`	補題	`notation`	記法	`answer`	解答

番号はこれらの環境で共有されていて，たとえば「定義 1.1，定理 1.2，注意 1.3，……」というようになります．`theorem*` 環境で，番号なしの定理を出すこともできます（`proposition*` 環境なども同様です）.

証明を分割して書くためのマクロとして，`subproof` 環境を定義しています．たとえば，

```
\begin{proof}
  \begin{subproof}{\IMPLIES{(a)}{(b)}}
    明らかである.
```

[*3] `\Re` と `\Im` はすでに LaTeX で定義されています（それぞれ数式モード内で $\Re$, $\Im$ を出力します）．再定義するのはやめておきました.

```
  \end{subproof}

  \begin{subproof}{\IMPLIES{(b)}{(a)}}
    先に述べた定理から従う.
  \end{subproof}
\end{proof}
```

と書けば，次のように出力されます[*4].

証明　(a) $\Longrightarrow$ (b)　明らかである.
(b) $\Longrightarrow$ (a)　先に述べた定理から従う. □

4.6 強調

\emph でテキストを強調できます．\emph は LaTeX で定義されているコマンドですが，tosuustd2019 はそれを和文用に再定義しています．tosuustd2019 の設定では，\emph で囲んだ部分について，和文はゴシック体に，欧文はサンセリフ体になります．たとえば，

```
$ 1 $ と自身以外に正の約数をもたない $ 2 $ 以上の整数を，\emph{素数（prime number）}という.
```

と入力すると，

1 と自身以外の正の約数をもたない 2 以上の整数を，素数（prime number）という.

と出力されます.

4.7 箇条書き

箇条書きの環境として，次の 15 個を定義しています.

環境名	ラベル	環境名	ラベル
enumarabic	1, 2, 3, ...	enumRoman	I, II, III, ...
enumarabicd	1., 2., 3., ...	enumRomand	I., II., III., ...
enumarabicp	(1), (2), (3), ...	enumRomanp	(I), (II), (III), ...
enumAlph	A, B, C, ...	enumroman	i, ii, iii, ...
enumAlphd	A., B., C., ...	enumromand	i., ii., iii., ...
enumAlphp	(A), (B), (C), ...	enumromanp	(i), (ii), (iii), ...
enumalph	a, b, c, ...		
enumalphd	a., b., c., ...		
enumalphp	(a), (b), (c), ...		

LaTeX 標準の enumerate 環境とは違い，これらの環境は，どのレベル（箇条書きのネストの深さ）に書いても同じラベルを出力します.

[*4] 2 つの subproof 環境の間に空行を挟まないと，適切に改段落されません.

4.8　参考文献リストの前に文章を挿入する

thebibliography 環境の前に \insertbeforebib{〈テキスト〉} と書くと，〈テキスト〉が参考文献の見出しとリストの間に挿入されます．

参考文献

ウェブページについては，2019 年 2 月 3 日にアクセスし，内容を確認しました．

[1] 奥村 晴彦，黒木 裕介，『[改訂第 7 版] LaTeX 2_ε 美文書作成入門』，技術評論社，2017.

[2] Will Robertson, “Every symbol (most symbols) defined by unicode-math”.
ftp://ftp.u-aizu.ac.jp/pub/tex/CTAN/macros/latex/contrib/unicode-math/unimath-symbols.pdf

[3] ZR，Qiita「LaTeX の「アレなデフォルト」 傾向と対策」.
https://qiita.com/zr_tex8r/items/297154ca924749e62471

[4] ZR，Qiita「UTF-8 で欧文文字を入力する 〜inputenc パッケージ〜」.
https://qiita.com/zr_tex8r/items/b40ca3478e4fe14868e5

[5] アセトアミノフェン，Acetaminophen’s diary「TeX Live 2018 注目ポイントまとめ (1)」.
http://acetaminophen.hatenablog.com/entry/tl2018-01

[6] TeX Wiki「hyperref」.
https://texwiki.texjp.org/?hyperref

[7] TeX Wiki「TikZ」.
https://texwiki.texjp.org/?TikZ