情報処理 2 第 2 回

TEX (2)

桂田 祐史 (かつらだ まさし)

2012 年 4 月 25 日

この授業用の WWW ページは http://www.math.meiji.ac.jp/~mk/syori2-2012/

1 連絡事項その他

- SNS への登録を希望するというメールを送ってくれた人については、登録は出来ているはずです (招待メールというのが届いているはずです)。
- これまで、SNS 登録の申請やレポートをメールで送ってくれたものの中に、かなりの割り合いのものに「不備」がありました。件名が書いていなかったり、メールアドレスが書いていなかったり。何をすべきか、授業中に話しているし、何より資料に書いてあります。こちらに**メールを送る前に、条件を満たしているか、確認する習慣**をつけて下さい。
- 次回は、(仮称) 十進 BASIC によるプログラミングをする予定です。

2 自分のパソコンへの TEX インストールのすすめ

TEX のようなソフトは「慣れ」が大きいので、積極的に使うことにするとご利益 (費用対効果) が大きいです。自分が自由に使えるパソコンがある場合、それに TEX をインストールしてみたらどうでしょう。特に数学の教員になるような人は、将来も有効に使える可能性があります。

情報処理教室で利用している TEX 関係のソフトのほとんどはフリーソフトです[1]。自分が自由に使えるパソコンがあり、高速なインターネット接続があれば、インストールすることが可能です (Windows だったら、あべのり「TeX インストーラ 3」[2] がお勧め)。また TEX に関する書籍には、インストール用の DVD-ROM がついている場合があるので (例えば奥村 [1])、それを利用するのも良いでしょう。

[1] ただし秀丸は TEX と直接関係はなく、学割制度のあるシェアウェアです。サクラエディタはフリーソフトです。

[2] http://www.math.sci.hokudai.ac.jp/~abenori/soft/abtexinst.html

3 前回のおさらい

3.1 TEX のための準備作業 (思い出すだけ)

- マイドキュメント (ライブラリ → ドキュメント) の下に授業用のフォルダー (“syori2” という名前の人が多いはず— 以下 syori2 として説明する) の中にコマンド・プロンプトへのショートカットを作り、その「作業フォルダ」をその授業用フォルダ (Z:¥.windows2000¥syori2 となる) にした。
  → 参考 http://www.math.meiji.ac.jp/~mk/syori2-2012/jouhousyori2-2012-01/node9.html

- LATEX 文書の雛形となる、tamago.tex という名前のファイルを syori2 の中に作成した。
  → 参考 http://www.math.meiji.ac.jp/~mk/syori2-2012/jouhousyori2-2012-01/node10.html

3.2 こうやって LATEX で文書を作る (やってみよう)

1. tamago.tex を (秀丸などのテキスト・エディターで) 開いて、[名前をつけて保存 (A)] で、適当な名前に変えて保存してから、編集 (執筆？) を始める。
   (試すなら、今日の課題 2 のために、kadai2.tex というファイルを作ることを勧める。)

2. なんとか.tex を dvi ファイルに変換 (コンパイル?) するには、コマンド・プロンプトで、

   ```
   Z:¥.windows2000¥syori2> platex なんとか.tex Enter
   ```

   とする (下線部を入力する, 以下繰り返さない)。

3. なんとか.dvi を表示 (プリビュー) するには、コマンド・プロンプトで

   ```
   Z:¥.windows2000¥syori2> dviout なんとか.dvi Enter
   ```

   とする。印刷は dviout のメニューから行う。

4. 人に渡す場合は、PDF ファイルにするとよい。なんとか.dvi を PDF ファイルに変換するには、コマンド・プロンプトで

   ```
   Z:¥.windows2000¥syori2> dvipdfmx なんとか.dvi Enter
   ```

   とする。

4 レポート課題2

TEX については、1年生のときに学んだはずだが、思い出してもらうために、次節以降 (5,6,7節) に書いた事項を試した kadai2.tex と言う TEX 文書を作り (tamago.tex を読み込んで別名保存してから書き始める)、SNS の「2012 年度情報処理 2」トピック「第 2 回」[3] を使って提出して下さい (提出時刻は、書き込み ボタンのクリックが 14:20 分以降とします)。

本文中に kadai2.tex の内容をコピー＆ペーストして、kadai2.pdf をファイルとして添付して下さい。締切は本日 4 月 25 日 (水) 18:00 とします (原則この授業時間内に提出してもらいますが、何か問題があったときのために 18:00 としておきます)。

5 LATEX 文書 .tex の書き方 — 入門

5.1 最初に覚えるべきこと

- ファイル名の拡張子は ".tex" とする。
- TEX のコマンドには、先頭に「バックスラッシュ」 \ をつけるが、日本語環境では、「円記号」 ¥ として表示される場合が多い。「円記号」 ¥ も「バックスラッシュ」 \ も、(見栄えは違うけれど、文字コードは同じなので[4]) TEX のコマンドにとっては同じである。
- パーセント記号 % から行末までは注釈になる。
- いつでも書くことになりそうな次の内容は、tamago.tex に書いておいた (自分の氏名などを書き足すと良いかもしれない)。

tamago.tex

```
\documentclass[12pt,leqno]{jarticle}
\begin{document}
\end{document}
```

これを読み込んで [別名で保存 (A)]、というやり方を勧める。

- タイトルをつけるには、

```
\title{はじめての \TeX}% タイトル
\author{桂田 祐史}      % 著者名
\date{2012 年 4 月 25 日}   % 日付（省略すると組版した日になる）
\maketitle               % これでタイトルを表示する
```

(date を省略すると、その時点の年月日が元号で表示されるが、西暦にするには、\ 西暦 コマンドを用いる。)

[3] http://sns.math.meiji.ac.jp/?m=pc&a=page_c_topic_detail&target_c_commu_topic_id=1374

[4] 最近、この「常識」が通用しなくなるケースも出て来ました。今の Mac では、購入時の状態で、¥ キーを押して入力されるのは、「バックスラッシュ」 \ とは異なる、Unicode で用意された「円記号」 ¥ です。日本語入力システム (たとえば「ことえり」) の設定を変更しないと「バックスラッシュ」 \ が入力できません。

```
\西暦
\title{はじめての \TeX}
\author{桂田 祐史}
\maketitle
```

- \begin{document} から \end{document} までの間に、ローマ字、数字などの “フツーの字” で書くとそのまま文書に入力される。いわゆる記号は注意が必要である。

細かい話: 記号について

まず、そのまま入力&表示できるものとして、

! " ’ ( ) - = ‘ @ [ ] + * ; : ? , .

がある (マイナス - は、1 文字の場合、2 文字連続の場合、3 文字連続の場合で、それぞれ - – — となるので、そのまま入力できるものと考えるべきではないかもしれない。もっとも通常、マイナスは数式中に現われるものだから「$-$ と書く」と覚えるべきかも。)。

一方、シャープ # などは、そのまま入力したのではダメで、これはバックスラッシュ \ を前につけて \# と入力する必要がある (\ でエスケープする、という)。同様にエスケープする必要がある文字としては、

$ % & _ { }

がある。

$ と $ で囲んで、数式モードで扱うべき文字としては

| < > −

がある。

難しいのは次の 3 文字で、これを表示するには、(右側に書いた) 専用のコマンドを用いなければいけない。

~ \textasciitilde
\ \textbackslash
^ \textasciicircum

日本人専用の応急処置として、難しい文字の入力には漢字を使う、という手がある (やや幅広になってしまうけれど)。# $ % &__ {}|＜＞〜＼という感じで簡単。

5.2 改行と空白 (最低限の注意)

意外と難しいので[5]、ある程度 TEX の説明が進んでから詳しく説明する。ここではごく基本的なことと、「予告」に止める。

[5]少し大げさなようだが、TEX の設計思想に係わることなので。

- .tex の中にいくら空白を続けても、一つ空白を入れたのと同じで、小さな空きができるだけ。

```
This is        a                                                    pen.
```

は

This is a pen.

となる。空白を明示的に入力するために、\quad などのコマンドがある。数式モードでは微調整用のコマンドがたくさんある。

- 連続した改行は「空行」と呼び、パラグラフ (段落?) を変更するという意味になり (\par と同じ)、改行されて、次の文の先頭に空きができる (いわゆる段落先頭の字下げ (indentation))。連続した空行は 1 つの空行と同じことになる。

- 英文中の一つの改行 (空行でないもの) は、一つの空白と同等。日本語文中の一つの改行 (空行でないもの) は、無視される。(不正確な言い方だが…)

```
I
love
you.
弁慶がな
ぎなたを
```

は

I love you. 弁慶がなぎなたを

となる。

- 強制的な改行は \\ だが、初心者が使いたくなるケースの 95% は誤用である (卒研で君達の先輩を相手にしたときの経験則)。

「(数式・表でないところで) 強制改行は極力使わない」

と考えることを勧める。

6 簡単な数式

6.1 数式モード

数式は「数式モード」の中で書く。数式モードには次の二つがある。

1. 文中の数式 (**インライン数式**) は、ドル記号 $ ではさんでかく。

```
ピタゴラスの定理から $a^2+b^2=c^2$ が成り立つ。
```

ピタゴラスの定理から $a^2+b^2=c^2$ が成り立つ。

2. 数式のみの行 (**ディスプレイ数式**) を作るには、色々な命令があるが、もっとも基本的なものは、\[と \] ではさむもので、例えば

```
ピタゴラスの定理から
\[
  a^2+b^2=c^2
\]
がなりたつ。
```

のようにすると

ピタゴラスの定理から
$$a^2+b^2=c^2$$
がなりたつ。

となる。式番号をつけるには equation 環境というものを用いて、

```
ピタゴラスの定理から
\begin{equation}
  a^2+b^2=c^2
\end{equation}
がなりたつ。
```

のように書く。最初に `\documentclass[12pt,leqno]{jarticle}` のように、leqno (left equation number) を指定してある場合は、式番号は左側につく。

ピタゴラスの定理から

$$a^2+b^2=c^2 \tag{1}$$

がなりたつ。

6.2 かっこ

丸い括弧 (,) とカギ括弧 [,] は普通に入力できる。{, } は前に \ をつける。

```
\[
  \{[(a+b)+c]+d\}
\]
```

とすると

$$\{[(a+b)+c]+d\}$$

となる。かっこの大きさを調節するには、\left と \right で挟む場合が多い。

```
\[
  \left[
   \left(x-x_0\right)^2+\left(y-y_0\right)^2
  \right]^{1/2}
\]
```

$$\left[\left(x-x_0\right)^2+\left(y-y_0\right)^2\right]^{1/2}$$

6.3 空白

数式モード中は、たくさんの空白用コマンドがある[6]。

```
\[
 a\,a\;a\ a\quad a\qquad a
\]
```

$$a\,a\;a\ a\quad a\qquad a$$

空白を詰めることも必要になる。\! で詰まる (マイナスの空白)。

```
\[
  \int\int f(x,y)dxdy=\int\!\!\int f(x,y)dxdy
\]
```

とすると

$$\int\int f(x,y)dxdy=\int\!\!\int f(x,y)dxdy$$

[6]quad (=quadrat) 印刷用語で空白の込め物 (広辞苑によると、「組版の際に、印刷する必要のない余白部を埋めるために組み込むもの。」だそうである。字と字の間に入れるのが「スペース」、大きな余白にいれるのが「クワタ」であるとか。) の一種。個人的には、焼き鳥の「ハツ」 (heart) を思い出してしまう…

となる (左辺と右辺の積分記号の間隔を比べよう)。

(もっとも最近の TEX には、重積分・三重積分用に、\dint, \tint というコマンドが用意されているので、出番は少なくなった ??)

6.4 色々な記号

6.4.1 ギリシャ文字

\ の後にローマ字 (ラテン文字) で読みを書くことでギリシャ文字が書ける。

```
\[
  \alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi
  % omicron は o と字の形が同じなのでない
  \pi\rho\sigma\tau\upsilon\phi\chi\psi\omega
\]
```

とすると

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

となる。

大文字のギリシャ文字は、先頭のローマ字を大文字にすればよい。例えば

```
\[
  \Gamma  \Delta  \Theta  \Lambda  \Xi  \Pi  \Sigma  \Upsilon  \Phi
  \Psi  \Omega
\]
```

とすると

$$\Gamma\Delta\Theta\Lambda\Xi\Pi\Sigma\Upsilon\Phi\Psi\Omega$$

となる (これ以外は、ローマ字の大文字と同じ。例えば α の大文字は A で良い。)。

なお、

```
\[
  \varepsilon\vartheta\varpi\varrho\varsigma\varphi
\]
```

とすると、

$$\varepsilon\vartheta\varpi\varrho\varsigma\varphi$$

6.4.2 集合

```
\[
  a\in A\subset B,\quad
  C\supset D,\quad
  a\not\in A,\quad
  C\not\supset D,\quad
  A\cup B, A\cap B, A\setminus B=\emptyset,\quad
  \bigcup_{i=1}^\infty A_i=\bigcap_{i=1}^\infty B_i
\]
```

$$a\in A\subset B,\quad C\supset D,\quad a\not\in A,\quad C\not\supset D,\quad A\cup B, A\cap B, A\setminus B=\emptyset,\quad \bigcup_{i=1}^\infty A_i=\bigcap_{i=1}^\infty B_i$$

$\in$ (\in) の逆向きが $\ni$ (\ni) であったり、$\subset$ (\subset) の逆向きが $\supset$ (\supset) であるのは、苦し紛れっぽいけれど…

6.5 上つき添字、下つき添字

a^2 は a^2 とする。a_n は a_n とする。2^{2^n} は 2^{2^n} とする。
積分やシグマなどもこの応用で、

```
\[
  \lim_{R\to\infty}\int_a^R f(x)dx=\sum_{n=1}^\infty a_n
\]
```

とすると

$$\lim_{R\to\infty}\int_a^R f(x)dx=\sum_{n=1}^\infty a_n$$

6.6 分数

```
\[
  \frac{a+b}{c}=\frac12
\]
```

は

$$\frac{a+b}{c}=\frac{1}{2}$$

となる。

分数や積分、和の記号など、インライン数式では小さく組版されるが、ディスプレイ数式と同じように大きく組版するには、\displaystle コマンドを用いる。

```
$\frac{a+b}{c}=\frac12$ は小さいので、
$\displaystyle\frac{a+b}{c}=\frac12$ とすると大きくなる。
```

は

$\frac{a+b}{c}=\frac{1}{2}$ は小さいので、$\displaystyle\frac{a+b}{c}=\frac{1}{2}$ とすると大きくなる。

実は \dfrac という命令もある (amsmath パッケージが必要)。

なお、¥displaystyle は長くて入力が面倒なので、後述するマクロなどを利用する人が多いようである。\begin{document} の前に

```
\newcommand{\dsp}{\displaystyle}
```

と定義しておくと、以下 \dsp で、\displaystyle としたのと同じになる。

6.7 sin などの「作用素」

$\boxed{\sin x}$ の 's', 'i', 'n' はイタリックでない[7]、いわゆる立体 (ローマン体) で、sin と x の間に適度のブランクがあることに注意。こういうものには、専用のコマンドが用意されている場合が多い。

```
\[
 \sin x=\log y=\max A
\]
```

$$\sin x=\log x=\max A$$

単に log x のようにと書くと $logx$ となってしまう (これでは l, o, g, x の積にしか見えない)。なぜだか考えてみることを勧める。

マクロというものを使って、自分でこの種のコマンドを作ることも出来る。\begin{document} の前に (「プリアンブルに」という)

```
\newcommand{\grad}{\mathop{\mathrm{grad}}\nolimits}
```

[7]普通、数式中のローマ字は、x のようにイタリック (斜めに傾いているのが特徴) で表すことに注意。

と書いておくと (呪文のようですが、“grad” のところだけ変えれば良い、と覚えましょう)、\grad というコマンドが定義できる。

6.8 矢印

```
\[
  \to \quad \mapsto \quad \leftarrow \quad \Leftarrow\quad
 \longleftarrow \quad \Longleftarrow
 \quad \leftrightarrow \quad \Leftrightarrow
 \quad \longleftrightarrow \quad \Longleftrightarrow
\]
```

とすると

$$\to \quad \mapsto \quad \leftarrow \quad \Leftarrow \quad \longleftarrow \quad \Longleftarrow \quad \leftrightarrow \quad \Leftrightarrow \quad \longleftrightarrow \quad \Longleftrightarrow$$

上下、斜めの矢印については、

```
\[
  \uparrow \quad \downarrow \quad \Uparrow \quad \Downarrow \quad
  \updownarrow \quad \Updownarrow \quad
  \nearrow \quad \nwarrow \quad \searrow \quad \swarrow
\]
```

とすると

$$\uparrow \quad \downarrow \quad \Uparrow \quad \Downarrow \quad \updownarrow \quad \Updownarrow \quad \nearrow \quad \nwarrow \quad \searrow \quad \swarrow$$

もちろん left の反対の right もある。
\to と \mapsto はそのまま覚え、それ以外は命名ルールを理解して覚えることを勧める。

6.9 点

```
\[
 \cdot \quad \cdots \quad \ldots \quad \ddots \quad \vdots
\]
```

は順に、真ん中に一つの点、真ん中に 3 つの点、下に 3 つの点、斜めに 3 つの点、垂直方向に 3 つの点となる (c は center, l は low, d は diagonal (対角線の), v は vertical (垂直の))。

$$\cdot \quad \cdots \quad \ldots \quad \ddots \quad \vdots$$

6.10 不等式

等号のつかないものはそのまま $<, >$ を使うとよい。$\leq$ は \le とし、$\geq$ は \ge とする[8]。

```
\[
 a<b\le c\ge d
\]
```

$$a < b \le c \ge d$$

なお \ll, \gg で $\ll, \gg$ となる。また、等号 $=$ の否定 $\neq$ は \ne と入力する。

平行線を含む $\leqq, \geqq$ は、それぞれ \leqq, \geqq と入力する (これは次の項で説明する AMS 拡張であるので、プリアンブルに \usepackage{amssymb} とする必要がある)。

6.11 その他の記号

$$\| \quad \pm \quad \mp \quad \times \quad \div \quad \sim \quad \simeq \quad \fallingdotseq \quad \leqq \quad \geqq \quad \nabla \quad \triangle \quad \partial \quad \forall \quad \exists \quad \infty \quad \propto \quad \angle \quad \langle \quad \rangle$$

を表示するには、次のようにする (\fallingdotseq $\fallingdotseq$ のような AMS (アメリカ数学会) 由来のフォントには、プリアンブルに \usepackage{amssymb} と書くことが必要です)。

```
...
\usepackage{amssymb}% AMS で用意したシンボルのフォント
...
\begin{document}
...
\[
  \| \quad \pm\quad \mp \quad \times \quad \div \quad
 \sim \quad \simeq \quad \fallingdotseq \quad \leqq \quad \geqq\quad
 \nabla \quad \triangle \quad
 \partial \quad
 \forall \quad \exists \quad
 \infty \quad \propto\quad
 \angle\quad \langle\quad \rangle
\]
...
```

ちなみに \fallingdotseq や \partial は長いので、筆者はマクロを使って短い別名を定義してある[9]。

[8] 多分、“less than or equal to” から le, “greater than or equal to” から ge となったのであろう。

[9] 解析屋にとって、偏微分記号 ∂ は良く使うので…

6.12 行列、ベクトル

行列や (縦) ベクトルでは、式 (成分) を「きれいに並べる」必要がある。このためには、array 環境や matrix 環境を用いる (縦ベクトルは、列の個数が 1 である行列とみなす)。また括弧 (と) (あるいは [,], {, }) は \left と \right を使って拡大する。

$$\left(\begin{array}{cc} a & b \\ c & d \end{array} \right) \left(\begin{array}{c} x \\ y \end{array} \right)$$

は

array 環境を用いて行列を書く

```
\[
 \left(
  \begin{array}{cc}
     a & b \\
     c & d
  \end{array}
 \right)
 \left(
  \begin{array}{c}
     x \\
     y
  \end{array}
 \right)
\]
```

または AMS 拡張に含まれる pmatrix 環境を用いても良い。

pmatrix 環境を用いて行列を書く

```
\documentclass[12pt,leqno]{jarticle}
...
\usepackage{amsmath}% プリアンブルに書く
...
\begin{document}
...
\[
  \begin{pmatrix}
     a & b \\
     c & d
  \end{pmatrix}
  \begin{pmatrix}
     x \\
     y
  \end{pmatrix}
\]
```

pmatrix 環境の方が使い方は簡単だが[10]、array 環境は左寄せ (l)、中央揃え (c)、右寄せ (r) など細かい制御ができる。

なお

$$|x| = \begin{cases} x & (x \geq 0 \text{ のとき}) \\ -x & (x < 0 \text{ のとき}) \end{cases}$$

も似た感じで出力できる (他に cases 環境というのもある)。

```
\[
 |x|=
 \left\{
 \begin{array}{rl}% 1 でなく l (エル L の小文字) left の頭文字なので
   x & \mbox{($x\ge 0$ のとき)}\\
  -x & \mbox{($x<0$ のとき)}
 \end{array}
 \right. % 右側は括弧なし (ドット . が重要)
\]
```

6.13 数式中の言葉

数式中に日本語や英語で説明の言葉を書きたいことがある。そういう場合は、\mbox{} や、\text{} を使う (後者は文字の大きさを回りの式に合わせて調節してくれる)。

[10] なお、括弧の形の違う行列を作る bmatrix, Bmatrix 環境、括弧なしの matrix 環境等もある。

```
\usepackage{amsmath}% \text{} に必要

...

\[
  f(x)=\log x\quad\mbox{($x$ は正の実数)},\quad
 \zeta(s)=\prod_{\text{$p$ は素数}}\frac{1}{1-\dfrac{1}{p^s}}
\]
```

$$f(x)=\log x \quad (x \text{ は正の実数}), \quad \zeta(s)=\prod_{p \text{ は素数}} \frac{1}{1-\dfrac{1}{p^s}}$$

6.14 数式の縦揃え

複数行に渡る数式を書く場合、等号など適当な位置で揃えたくなることがある。色々なやり方があるが、とりあえず align, align* 環境を覚えておきましょう。

等号の位置を揃える

```
\begin{align*}
 x&=1,\\
 f(x)&=10.
\end{align*}
```

$$\begin{aligned} x&=1,\\ f(x)&=10. \end{aligned}$$

行の先頭を揃える

```
\begin{align*}
 &x=1,\\
 &f(x)=10.
\end{align*}
```

$$\begin{aligned} &x=1,\\ &f(x)=10. \end{aligned}$$

アスタリスク (*) なしの align 環境は数式番号がつく。

等号の位置を揃える (数式番号つき)

```
\begin{align}
 x&=1,\\
 f(x)&=10.
\end{align}
```

$$x = 1, \tag{2}$$

$$f(x) = 10. \tag{3}$$

(ちなみに単独の式で式番号をつけるには、\[と \] の代りに \begin{equation} と \end{equation} (equation 環境) を用いる。

```
\begin{equation}
  3^2+4^2=5^2.
\end{equation}
```

$$3^2 + 4^2 = 5^2. \tag{4}$$

6.15 書き足すべきこと

- 数式番号の参照　\label{} と \ref{}
- 目次を作る　\tableofcontents
- 索引の作り方

7 TEX のマクロ機能、パッケージ機能の紹介

7.1 マクロ

既に紹介したように、プリアンブルに

gradient 作用素の記号を定義する

```
\newcommand{\grad}{\mathop{\mathrm{grad}}\nolimits}
```

と書いておくと、\grad というコマンドが定義できる。これは TEX のマクロという機能を使っている。

マクロは、簡単な部分だけでも、便利に使うことが出来る。例えば \displaystyle コマンドのように、長くて入力が面倒なコマンドに、短い別名をつけるために使うことが出来る。そのためには、プリアンブルに例えば

\displaystyle を手短に \dsp で

```
\newcommand{\dsp}{\displaystyle}
```

のように書けば良い。

マクロでは、いわゆる引数を用いることができる。2×2 の行列 $\left(\begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right)$ は、例えば

```
\left(
\begin{array}{cc}
 1 & 2 \\
 3 & 4
\end{array}
\right)
```

として組版できるが、

2×2 行列用のマクロ

```
\newcommand{\gyouretsu}[4]{
 \left(
 \begin{array}{cc}
  {#1} & {#2}\\
  {#3} & {#4}
 \end{array}
 \right)
}
```

とマクロ \gyouretsu を定義しておくと (行列の 4 つの成分が引数として与えられる)、

```
\gyouretsu{1}{2}{3}{4}+\gyouretsu{5}{6}{7}{8}=
\gyouretsu{6}{8}{10}{12}
```

で $\left(\begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right) + \left(\begin{array}{cc} 5 & 6 \\ 7 & 8 \end{array} \right) = \left(\begin{array}{cc} 6 & 8 \\ 10 & 12 \end{array} \right)$ が組版できる。

なお、マクロの名前には、ローマ字のみが使えます (gyouretu22 のような文字列は使えません)。

実は、通常使っている LaTeX そのものが、膨大なマクロの集成に他なりません。

7.2 パッケージ

LaTeX である程度まとまったことをやりたい場合に、パッケージというものが用意されていることがあります (中身は要するにマクロの集合です)。

パッケージは、プリアンブルで \usepackage{} コマンドを用いて読込みます。

詳細は省略しますが (自分で必要になってから調べれば良い)、以下筆者が良く利用しているものの名前をあげておきます。

geometry パッケージ TeX 文書で使う**紙の大きさや、余白の長さ**などを指定するのに、ge-

ometry パッケージ[11] というものが便利である。

```
\usepackage[a4paper]{geometry}
```

のように使う。

amsmath, amssymb パッケージ 複雑な数式や、やや珍しい記号類の組版には、アメリカ数学会 (American Mathematical Society, AMS) が開発した amsmath, amssymb パッケージが威力を発揮する。

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

graphicx パッケージ グラフィックスを取り込むための \includegraphics{} 命令が用意されている (使い方は後述する)。

```
\usepackage[dvips]{graphicx}
```

LaTeX Beamer パッケージ プレンゼンテーション資料を TEX で作るために、色々なパッケージが開発されている。LaTeX Beamer パッケージはその一つである。このあたりは流行り廃りがあるので、自分が必要になったときに、WWW で検索すると良い (例えば「外圏 Wiki カテゴリ:LaTeX」[12])

ascmac パッケージ 円記号を組版する \yen や、枠で囲う screen 環境、見出しつきの枠で囲う itembox 環境などは、ascmac パッケージにある。

8 LATEX 文書への外部からのファイルの取り込み

(この節の内容は、実際に必要になってから読むのがお勧めです。)

例えばプログラミングがらみの課題のレポートを作る場合など、ソースプログラムや画像イメージなどを取り込みたくなります。

8.1 ソースプログラム等テキストファイルの取り込み

短いものは

[11] http://tug.ctan.org/tex-archive/macros/latex/contrib/geometry/
[12] http://windom.phys.hirosaki-u.ac.jp/fswiki/wiki.cgi?action=CATEGORY&category=L'aTeX

verbatim 環境の利用

```
\begin{verbatim}
#include <stdio.h>
int main(void)
{
  printf("Hello, world\n");
  return 0;
}
\end{verbatim}
```

のように、.tex ファイルの中の、verbatim (“verbatim” は「言葉通りに」、「逐語的に」という意味の単語) 環境の中に入れてしまえばよいですが、長いものや頻繁に変更を加えるものを扱うのは面倒です。verbatimfiles パッケージを組み込むと有効になる \verbatimfile コマンドや \verbatimlisting コマンド (行番号つき) を使うとよいでしょう。

hello.c を取り込む

```
\documentclass[12pt,leqno]{jarticle}
\usepackage{verbatimfiles}% パッケージを組み込む（複数形の s がついている）

\begin{document}
...
\verbatimfile{hello.c}% hello.c は別途用意してあるとして
...
\end{document}
```

verbatimfiles.sty というファイルが必要ですが、例えば http://www.math.meiji.ac.jp/~mk/labo/tex/style/verbatimfiles.sty から入手して、.tex ファイルと同じフォルダ (ドキュメントの下にある syori2 という人が多いはず) に置いて下さい。具体的には、マウスカーソルをリンクに合わせて、マウスを右クリックして、「名前をつけてリンク先を保存」あるいは「対象をファイルに保存 (A)」を選択します (Internet Explorer では、普通に表示した後に、[ファイル] メニューから [名前をつけて保存] で保存するには、[テキストファイル (*.txt)] 形式を選択して保存し、保存がすんでから verbatimfiles.sty という名前に変更する必要があります。面倒で間違えやすいので、ここでは右クリックして保存する方法を推奨します。)

8.2 画像の取り込み

画像ファイルには色々なフォーマットがあるが、LaTeX に取り込むには、**カプセル化 PostScript 形式** (Encapsulated PostScript, 長いので EPS 形式と呼ぶことにする, 通常は “.eps” という拡張子をつける) に変換してから、\includegraphics 命令で取り込むのが簡単で問題が生じにくい。

十進 BASIC のグラフィックスの場合、「名前をつけて保存 (A)」から JPEG 形式で (ファイ

ル名拡張子は ".JPG") 保存しておき、**jpeg2ps**[13] コマンドで EPS 形式に変換するのが便利である。

2012 年度の情報処理教室の Windows 環境には、Cygwin がインストールされていて、その中に jpeg2ps が入っている。コマンドプロンプトや、Cygwin のシェルで

```
Z:¥.windows2000¥syori2>jpeg2ps kamehosi.JPG > kamehosi.eps
```

とすると、kamehosi.JPG を EPS 形式に変換したファイル kamehosi.eps が出来る。

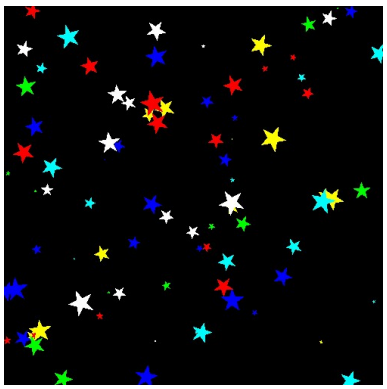
なお、Windows 7 の GUI で使える wjpeg2ps[14] というプログラムもある。使い方は簡単で、JPEG ファイルを wjpeg2ps のアイコンにドラッグして、convert ボタンを押すだけで EPS 形式のファイルが出来る。

仕組みについて、もう少し詳しい説明が読みたければ、「イメージデータの TEX への取り込み — jpeg2ps のすすめ」[15] を見てください。

kamehoshi2.eps を取り込む

```
\documentclass[12pt,leqno]{jarticle}
\usepackage[dviout]{graphicx}% graphicx パッケージが必要

\begin{document}
\begin{center}
 \includegraphics[width=5cm]{kamehoshi2.eps}% データは各自用意してください。
\end{center}
\end{document}
```



dviout でカラー表示・印刷をするには

カラーで表示・印刷するには、dviout で Option → Setup Parameters → Graphic で、GIF の取り扱いの設定で BMP(full-color) を選択する。dviout 起動時に、-GIF=5 というオプ

[13] http://www.pdflib.com/
[14] http://www.vector.co.jp/soft/dl/win95/art/se248407.html
[15] http://www.math.meiji.ac.jp/~mk/labo/howto/jpeg2ps.html

ション引数を指定しても良い。これをデフォールトの設定にする人も多いが、情報処理教室のプリンターはモノクロなので、さぼってある。

余談: ウィンドウの画像を取り込む

Windows 7 のウィンドウの画像をファイルに保存したければ、マウスカーソルを取り込みたいウィンドウに置いて、キーボードから Alt+Print Screen (Print Screen は、場合によっては Fn キーと一緒に押す必要があり、その場合は Alt+Fn+PrintScreen となる) を入力し、**ペイント**[16]のようなソフトにペーストしてから、適当に編集した後で、保存すると良いでしょう (もちろん JPEG 形式に出来る)。

(目次・章・節・索引などの文書の構造がらみのもの、定理環境、式番号の扱い、プレゼンテーション資料の作り方等はもう少し後で。急いでいる人は、奥村 [1] のような書籍を購入したり、ネットで調べたり、あるいは SNS の 2012 年度情報処理 2 コミュニティの「質問コーナー」[17]で質問して下さい。)

A ¥と \ の関係は ??

分かってしまうと簡単なことなのですが、手短かな説明で分かってもらうのは案外難しい「¥と \ 問題」です。

乱暴に言い切ってしまうと、¥の JIS コードも \ の ASCII コードも、どちらも 92 で、コンピューターのプログラム (Windows や TEX の内部処理) にとっては、同じものである、ということです。

コンピューターの中では、どういうデータも数列として表現されます。例えば “I am a boy.” は、私が今使っているコンピューターでは、

$$73, 32, 97, 109, 32, 97, 32, 98, 111, 121, 46$$

という数列になります。いきなり分かりづらいと思うかも知れませんが、落ち着いてみると、空白も 1 文字と考えると、“I am a boy.” は 11 文字からなる文で、数列の方は 11 項からなる数列なので、1 文字が 1 つの数に対応しているのだろう、そう考えてみると、空白は 32 という数で表わすらしい、きっと

$$\mathrm{I} = 73, \mathrm{a} = 97, \mathrm{b} = 98, \mathrm{m} = 109, \mathrm{o} = 111, \mathrm{y} = 121, . = 46$$

だろう、まで分かります。

実は ASCII というアメリカの規格では、バックスラッシュは 92 (16 進数では 5c) という数で表わすことになっています。

ASCII コード表

[16] スタート → すべてのプログラム (P) → アクセサリ → ペイント として起動できる。

[17] http://sns.math.meiji.ac.jp/?m=pc&a=page_c_topic_detail&target_c_commu_topic_id=1375

```
0x20 ( 32):      0x21 ( 33): !    0x22 ( 34): "    0x23 ( 35): #
0x24 ( 36): $    0x25 ( 37): %    0x26 ( 38): &    0x27 ( 39): ’
0x28 ( 40): (    0x29 ( 41): )    0x2a ( 42): *    0x2b ( 43): +
0x2c ( 44): ,    0x2d ( 45): -    0x2e ( 46): .    0x2f ( 47): /
0x30 ( 48): 0    0x31 ( 49): 1    0x32 ( 50): 2    0x33 ( 51): 3
0x34 ( 52): 4    0x35 ( 53): 5    0x36 ( 54): 6    0x37 ( 55): 7
0x38 ( 56): 8    0x39 ( 57): 9    0x3a ( 58): :    0x3b ( 59): ;
0x3c ( 60): <    0x3d ( 61): =    0x3e ( 62): >    0x3f ( 63): ?
0x40 ( 64): @    0x41 ( 65): A    0x42 ( 66): B    0x43 ( 67): C
0x44 ( 68): D    0x45 ( 69): E    0x46 ( 70): F    0x47 ( 71): G
0x48 ( 72): H    0x49 ( 73): I    0x4a ( 74): J    0x4b ( 75): K
0x4c ( 76): L    0x4d ( 77): M    0x4e ( 78): N    0x4f ( 79): O
0x50 ( 80): P    0x51 ( 81): Q    0x52 ( 82): R    0x53 ( 83): S
0x54 ( 84): T    0x55 ( 85): U    0x56 ( 86): V    0x57 ( 87): W
0x58 ( 88): X    0x59 ( 89): Y    0x5a ( 90): Z    0x5b ( 91): [
0x5c ( 92): \    0x5d ( 93): ]    0x5e ( 94): ^    0x5f ( 95): _
0x60 ( 96): ‘    0x61 ( 97): a    0x62 ( 98): b    0x63 ( 99): c
0x64 (100): d    0x65 (101): e    0x66 (102): f    0x67 (103): g
0x68 (104): h    0x69 (105): i    0x6a (106): j    0x6b (107): k
0x6c (108): l    0x6d (109): m    0x6e (110): n    0x6f (111): o
0x70 (112): p    0x71 (113): q    0x72 (114): r    0x73 (115): s
0x74 (116): t    0x75 (117): u    0x76 (118): v    0x77 (119): w
0x78 (120): x    0x79 (121): y    0x7a (122): z    0x7b (123): {
0x7c (124): |    0x7d (125): }    0x7e (126): ~
```

さて、これを日本に持って来て JIS 規格にする際に、ほんのちょびっと (2ヶ所) だけ、変更を加えました。

そのうちの一つが、92 を (バックスラッシュでなく) 円記号 ¥を表わすことにしたことです。つまり、通貨記号 ¥を含めたいので、比較的利用頻度が低いバックスラッシュのコードを使ってやろう、ということです。

参考文献

[1] 奥村晴彦, LaTeX 2_ε 美文書作成入門 改訂第 5 版, 技術評論社 (2010).

[2] 日本語 TeX 情報, http://oku.edu.mie-u.ac.jp/~okumura/texfaq/