

第 3 章

L^AT_EX

3.1 はじめに

3.1.1 T_EX (テフ, テック), L^AT_EX (ラテフ, ラテック, etc) とは

T_EX とは、数式の処理に優れる組版ソフトウェアであり、その T_EX を使ってもっと簡単に論文やレポートを作成出来るように開発されたのが L^AT_EX や L^AT_EX 2_ε である。

HTML 文章と同じように、命令を用いてページを作成していく。沢山の命令があるが、基本的なものさえ覚えておけば、後は必要なときに調べればよい。

例 3.1.1. (L^AT_EX ソースファイル)

TeX ソース (1)

```
%documentclass[dvipdfmx,a4j,12pt]{jarticle}
%ここがプリアンブル
\begin{document}
\textbf{第 1 回 集合の基礎、数学で使う記号}
\begin{flushright}
学生番号 \underline{\hspace{30mm}} 氏名 \underline{\hspace{40mm}}
\end{flushright}

\textbf{問題 1-1.} 以下の集合 $A, B, C$ に対して、(1) から (4) に答えよ。

(略)

\end{document}
```

T_EX では細かい設定を

プリアンブル (`\documentclass` と `\begin{document}` の間)

に指定出来る。

例 3.1.2. (例 3.1.1 を変換 (コンパイル) して表示した例.)

コンパイル (1) —

第 1 回 集合の基礎、数学で使う記号

学生番号_____ 氏名_____

問題 1-1. 以下の集合 A, B, C に対して、(1) から (4) に答えよ。

(略)

最初はプリアンプルの指定は気にせずに、慣れるまでは

`\begin{document}` と `\end{document}` の間

の部分のみを編集していく。

3.1.2 実際に使用してみる

T_EX は話を聞くより実際に使用した方が習得出来る。(細かいことは後述。)

T_EX では作成したソースファイルを DVI ファイルや PDF ファイルに変換して、初めて文章、数式や表などが表示される。この講義 (実習) では、TeXworks というランチャー兼エディタを使用して T_EX のファイル作成、PDF の表示を行う。

デスクトップ上の **TeXworks** を起動すると、黒い画面 (コマンドプロンプト) が表示されるが、少し待つと起動する。何も入力されていない新規のファイルが表示されたら、

TeX ソース (2) —

```
\documentclass[dvipdfmx,a4j,12pt]{jarticle}
\begin{document}
吾輩は猫である。名前は真田奈衣。

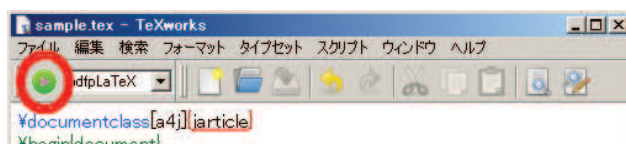
どこで生れたかとうんを見当がつかぬ。
何でも薄暗いじめじめした所で
ニャーニャー泣いていた事だけは記憶している。
吾輩はここで始めて人間というものを見た。
\end{document}
```

と入力する。入力が終わったら、[ファイル] から [名前を付けて保存] を選び、適切なフォルダを選択する。フォルダを選んだら、ファイル名を入力する。

その際、ファイル名 (フォルダ名) は 日本語, 空白入り, 記号 は避けた方が良いので^{*1}、`sample.tex` や `HandM01.tex` などとする。

^{*1} 使用可能なものと不可能なものがある。なので使わない方がよい

[保存] 後、左上のアイコン (下図の丸で囲っているアイコン) をクリックする。



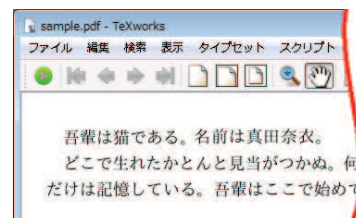
アイコンが  に変わり、少し待ち、エラーが無ければ元に戻り、右側に pdf ファイルが表示される。

アイコンが  のままならエラーメッセージが“ログの表示”に表示されている。

エラーの場合は、 をクリックして変換を中止し、エラーメッセージを元に修正し、再度アイコンをクリックする。

よく見かけるミスは、全角と半角の入力ミスと $\$$, $\{$, $\}$ の数が対応していないミス。

例えば、 $\$$ で囲むべき $\$y=x^2+3x\$$ が $\$y=x^2+3x$ や $\$y=x^2+3x \$$ となっていた場合、



```
! Missing $ inserted.
<inserted text>
          $
1.13
```

のようなエラーメッセージが“ログの表示”に表示される。

3.1.3 $\$begin\{document\}$ と $\$end\{document\}$ の間

★ 文字入力

(単純に入力した) 日本語や English はそのまま表示される。ただし、機種依存文字は使えない。

例えば、全角入力した I は表示されない。 I は半角大文字の I を使う。

また、太文字 や *italic type* は命令が必要で、太文字なら $\$textbf\{太文字\}$ 、イタリック体なら $\$textit\{italic type\}$ と入力する。

★ 記号入力

そのままでは入力できない記号もある。例えば、 $\$$ や $\&$ などは命令文に使われるため、表示には一手間かかる。

★ 下線の入力

下線は $\$underline\{下線\}$ と入力する。さらに、 $\$underline\{\$underline\{下線\}\}$ と入力すれば、下線となる。

★ 改行

改行も一工夫必要で、例えば L^AT_EX ソースファイル内に

TeX ソース (3) —————
岡山県岡山市北区理大町 1 - 1
岡山理科大学

と書いても

コンパイル (3) —————
岡山県岡山市北区理大町 1 - 1 岡山理科大学

と表示される。改行したい場合は、空行を入れる。したがって、

TeX ソース (4) —————
岡山県岡山市北区理大町 1 - 1

岡山理科大学

とすると改行され

コンパイル (4) —————
岡山県岡山市北区理大町 1 - 1

岡山理科大学

となる。ただし、1 つ以上の改行を入れても空行は出来ない。そのため、空行を広げたい場合は `\vspace` を使う。その際、`\vspace` の前に空行を挟むこと。

TeX ソース (5) —————
岡山県岡山市北区理大町 1 - 1

`\vspace{2cm}`
岡山理科大学

コンパイル (5) —————
岡山県岡山市北区理大町 1 - 1

岡山理科大学

★ 空白

空白を入れるには半角スペースを入れれば1つ空白が出来る。ただし、1つ以上入れても1つの空白しか入らない。例えば、半角スペースを複数入力して

TeX ソース (6) —————
 学生番号 (), 氏名 ()

と入力しても、変換結果は

コンパイル (6) —————
 学生番号 (), 氏名 ()

と表示される。

1つの空白以上の空白が必要な場合は `\hspace` を使う。例えば、`\hspace{2cm}` は 2cm の空白 () をあけることを意味する。

また、改行後に空白が入る (この行の先頭のように) ことを避けたい場合は `\noindent` を使う。(この場合、`\noindent` の前に空行を挟むこと。)

例 3.1.3. (3行前の文章の改行後に空白を入れない例)

TeX ソース (7) —————
 の空白 (`\hspace{2cm}`) をあけることを意味する。

`\noindent`
 また、改行後に空白が入る (この行の先頭のように) ことを避けたい場合は

とすると、

コンパイル (7) —————
 の空白 () をあけることを意味する。
 また、改行後に空白が入る (この行の先頭のように) ことを避けたい場合は

のように表示される。

☆ 数式 I

数式は数式入力モードで入力する必要があり、また数式入力モードには、インラインモードとディスプレイモードがある。インラインモードは文中に数式を書く場合に用い、ディスプレイモードは別立て (1行立て) で数式を書く場合に用いる。

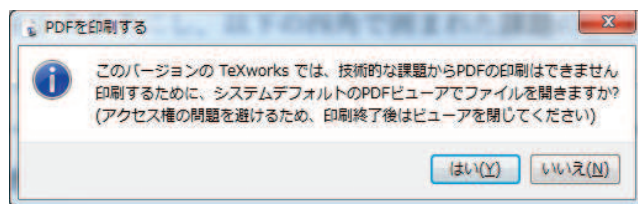
インラインモードは、数式のコマンド等を $\$ \$$ や $\mathbb{Z}$ ($\mathbb{Z}$) (など) で囲む。

数式の変数などはそのまま x, y と入力するとフォントが文中のものになるので、数式を表す命令 ($\$x, y\$$) を使うことによって、 x, y と表示される。

3.1.4 印刷

出来た PDF ファイルを印刷する場合は、PDF ファイルが表示されている(右側の) ウィンドウの [ファイル] から、[PDF を印刷する] を選ぶ。

そのときに、右図のようなメッセージが出るので [OK] を押す。



その後、PDF ビューアー (Adobe Acrobat Reader) が開かれるので、[ファイル] から [印刷] を選択し印刷を行う。

課題 3.1.1. 例 3.1.1 を参考にし、以下の四角で囲まれた課題の中身が表示されるような $\text{\LaTeX}$ ファイルを作成せよ。なお、連分数にある $\cdots$ は、`\cdots` を使うこと。

また、改行位置については気にしないでよい。

課題

第 2 回 数式, 図形の表現 I

学生番号 _____ 氏名 _____ 得点 _____

問題 1-1. 整式 $P(x) = ax^3 + bx^2 - 8x - 7$ が $x + 1$ で割り切れ、 $x - 2$ で割ったときの余りが -3 となるように、定数 a, b の値を定めよ。

問題 1-2. 次の等式を証明せよ。

$$\frac{1}{1-x} + \frac{1}{1-y} = 1 + \frac{1-xy}{(1-x)(1-y)}$$

問題 1-3. 円周率 π の連分数展開は以下であることを示せ。

$$\pi = 3 + \frac{1}{7 + \frac{1}{15 + \cdots}}$$

♠ **補足** 別ファイルで作成した $\text{\LaTeX}$ のソースファイルを読み込むことも出来る。たとえば、`sin.tex` を別ファイルで作成し、読み込むには

```
\input{filename.tex}
```

とする。命令文の行数の多い図や表など もしくは、何度も使うような命令文は、別のファイルに保存しておくと便利である。

3.2 T_EX コマンド

3.2.1 T_EX のコマンドと注意事項

★ 章, 節

文章を書くとき、章の見出し、節の見出し、本文 ... のようなレイアウトで書く。T_EX では章や節を `\section{***}` や `\subsection{***}` で記述することによって 左上のように、上下のスペース、フォントサイズの調整と番号付けを行ってくれる。これは、章や節の追加や削除を行ったとき、自分で番号の付け替えを行わなくて良いことを意味する。

★ 注釈

% 以降の記述は注釈として解釈され、本文に影響を与えない。

★ 地の文と命令文

T_EX の命令文は `\` で始まることが多い。たとえば、L^AT_EX と表示するには、`\LaTeX` と入力している。このとき、`\LaTeX` は命令文で、それ以外は地の文である。これを、

“このとき、`\LaTeX` は命令文で、”

と続けて書くと、`\` 以降どこまでが命令文で、どこからが地の文か解らなくなる。(特に英文ならなおさら。) そこで、命令文の終わりにはスペースを 1 つ以上入れるか、または命令文を { } で囲まなければならない。

♡ 注意! 後ろに空白を必要としない命令もある。例えば、`\$` や `\&` など。これらは空白を入れると \$ 30 のようにスペースが出来て不格好になる。

♠ 補足 数字やスペースは全角でなく半角を使う (方が望ましい)。

“問題 1. ”と“問題 1.”では、どちらが好みか。

3.2.2 パッケージとプリアンブル

今後必要となる以下のパッケージをプリアンブルに指定する。

```
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{arydshln}
\usepackage{ascmac}
```

プリアンブルは、`\documentclass` と `\begin{document}` の間 (例 3.1.1 参照)。

これらパッケージを読み込んでおくことによって、使えるコマンドが増える。これら以外にも多数のパッケージが存在するが、同時に使えないパッケージの組み合わせなどもあるので注意が必要である。

3.2.3 TeX の命令と種類

単独で命令となるものと、引数を伴い命令となるもの

例えば、先の `\LaTeX` や数式モード内の命令 `x^2`, `\cdots` などは単独で命令となっている。一方、分数を表す `\frac{ }{ }` や下線を引く `\underline{ }` などは `{ }` 内にある引数に対して命令を行う。

`\begin` と `\end` で命令となるもの

例 3.1.1 にあった、右寄せの `\begin{flushright} ~ \end{flushright}` や、中央寄せの `\begin{center} ~ \end{center}` などがそうである。

☆ 数式 II

`$$` と `$$$` で囲む以外のディスプレイモードとして `align` を紹介する。他にも、`equation`, `flalign`, `gather` などなどがあるがここでは省略する。必要に応じて調べてみてほしい。

この `align` は、`\begin{align} ~ \end{align}` で使用する。例えば、

TeX ソース (10) —

```
\begin{align}
y&=x^2 - 4x + 3\qquad \% =の前に & を入れて、以降の=を揃える。
&=(x-3)(x-1)
\end{align}
```

とすると、

コンパイル (10) —

$$y = x^2 - 4x + 3 \quad (1)$$

$$= (x - 3)(x - 1) \quad (2)$$

となる。各式の右側に付く式番号は自動的に通し番号になる。式番号をつけたくない場合は、その行の終わりに `\notag` をつける。全ての行に式番号を付けない場合は、`{align*}` に変える。

TeX ソース (11) —

```
\begin{align}
y&=x^2 - 4x + 3\notag\qquad
&=(x-3)(x-1)
\end{align}
```

コンパイル (11) —

$$y = x^2 - 4x + 3$$

$$= (x - 3)(x - 1) \quad (3)$$

☆ カッコ

カッコのうち、 $()$, $[]$ はそのまま表示される。 $\{ \}$ は `\{ \}` と入力する。

$(\frac{\sqrt{2}}{2}, 1)$ のようなとき、 $()$ の高さを調節するにはそれぞれの $()$ の前に `\left` または `\right` を付けくわえて `\left( \right)` と入力すると、 $(\frac{\sqrt{2}}{2}, 1)$ のように調整してくれる。

☆ 三角関数, 指数関数, 対数関数, n 乗根

$\sin, \cos$ は `\sin, \cos` と書き、 $\exp$ も `\exp` と書く。このとき `\` を忘れると *sin* や *exp* となる。 e^x と書く場合は、 e の書体を e とするため、`\mathrm{e}^{\mathrm{x}}` と書く。

平方根は `\sqrt[n]{x}` を使う。例えば、 $\sqrt{x^2 + y^2}$ は `\sqrt{x^2 + y^2}` と書き、 n 乗根は `\sqrt[n]{x}` を使い、 $\sqrt[3]{-8}$ は `\sqrt[3]{-8}` と書く。

☆ 総和, 総乗

総和と総乗はそれぞれ `\sum_{k=1}^n`, `\prod_{k=1}^n` を使う。

たとえば、 $c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$ は `c_{ij}=\sum_{k=1}^na_{ik}b_{kj}` と書く。同じ命令文でも `$$$` で囲めば文中 (数式モード) と異なり、以下のような表示となる。

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

☆ 極限, 微分, 積分

極限は `\lim_{x \rightarrow 0} f(x)` と書くと、 $\lim_{x \rightarrow 0} f(x)$ と表示される。微分は $f'(x)$ と入力して $f'(x)$ と表示したり、分数を用いて `\frac{dy}{dx}` と入力することによって $\frac{dy}{dx}$ と表示させたりする。

積分は `\int_{\text{下付}}^{\text{上付}} f(x) dx` を使う。 $\int_0^1 f(x) dx$ ならば、`\int_{0}^{1} f(x) dx` と入力する。これらは、数式立てにすると

$$\lim_{x \rightarrow 0} f(x), \frac{dy}{dx}, \int_0^1 f(x) dx$$

と表示される。ちなみに、 ∞ は `\infty` と書く。

☆ ベクトル, 行列

ベクトルは `\vec{a}` を使い、`\vec{a}` と書くと $\vec{a}$ と表示される。ただし、文字が多いときは $\overrightarrow{AB}$ では寂しいので、`\overrightarrow{AB}` と書いて $\overrightarrow{AB}$ と表示させる。

行列の表示にはいくつかの方法があるが、汎用性の高い `array` 環境を使う^{*2}。使用 방법은以下の例を参考するとよい。

^{*2} この `array` 環境も数式モード、数式立てで使う。

array 環境使用例

出力	コマンド	コメント
$\begin{array}{cc} a & b \\ c & d \end{array}$	<code>\left\begin{array}{cc}</code> <code>a & b \\\</code> <code>c & d</code> <code>\end{array}\right\ </code>	1 行目の{cc}の 2 つの c は 中央 (<i>center</i>) で 2 列の配列を意味する。 \\ は改行、行の終わり
$\left(\begin{array}{cc c} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{array}\right)$	<code>\left(\begin{array}{cc:c}</code> <code>a_{11} & a_{12} & b_1 \\\</code> <code>a_{21} & a_{22} & b_2</code> <code>\end{array}\right)</code>	{cc:c} は 2 列目と 3 列目の間に点線を入 れることを表す。
$\begin{array}{ c cc } \hline & a & b \\ \hline c & ac & bc \\ de & ade & bde \end{array}$	<code>\begin{array}{ c rl }</code> <code>& a & b \\\hline</code> <code>c & ac & bc \\\</code> <code>de & ade & bde</code> <code>\end{array}</code>	{ c rl } は 2 列目と 3 列目の間以外に縦線 を入れることを表し、 1 列目は中央, 2 列目は 右, 3 列目は左に揃えて いる。\\hline は横線を入 れる。
$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$	<code>\left[\begin{array}{ccc}</code> <code>a_{11} & \cdots & a_{1n} \\\</code> <code>\vdots & \ddots & \vdots \\\</code> <code>a_{n1} & \cdots & a_{nn}</code> <code>\end{array}\right]</code>	$\cdots$, $\vdots$, $\ddots$ の出力は <code>\cdots</code> , <code>\vdots</code> , <code>\ddots</code> を使う。

◇ 余談 行列や行列式の表示は pmatrix や vmatrix を使うことも多い。汎用性はない。

★ 枠, 表

表の作成には array 環境以外に table 環境がある。こちらの使用は数式環境ではない。

```

\begin{table}[htb]
\begin{tabular}{|l|c|c|l|} \hline
単位 & 評価 & 点数 & GPA \\ \hline
認定 & S & 100~90 & 4 \\
      & A & 89~80 & 3 \\
      & B & 79~70 & 2 \\
      & C & 69~60 & 1 \\ \hline
不認定 & D & 59~0 & 0 \\
      & E & 未資格 & 0 \\ \hline
\end{tabular}
\end{table}

```

単位	評価	点数	GPA
認定	S	100~90	4
	A	89~80	3
	B	79~70	2
	C	69~60	1
不認定	D	59~0	0
	E	未資格	0

ここではこの table 環境について 紹介のみに留め、詳しく触れないことにする。

☆ その他 (ギリシャ文字)

ギリシャ文字は、数式モード、数式立てで使用するここと。

大文字	コマンド	小文字	コマンド	大文字	コマンド	小文字	コマンド
A	<code>A</code>	α	<code>\alpha</code>	N	<code>N</code>	ν	<code>\nu</code>
B	<code>B</code>	β	<code>\beta</code>	Ξ	<code>\Xi</code>	ξ	<code>\xi</code>
Γ	<code>\Gamma</code>	γ	<code>\gamma</code>	O	<code>O</code>	o	<code>o</code>
Δ	<code>\Delta</code>	δ	<code>\delta</code>	Π	<code>\Pi</code>	π	<code>\pi</code>
E	<code>E</code>	ϵ	<code>\epsilon</code>	P	<code>P</code>	ρ	<code>\rho</code>
Z	<code>Z</code>	ζ	<code>\zeta</code>	Σ	<code>\Sigma</code>	σ	<code>\sigma</code>
H	<code>H</code>	η	<code>\eta</code>	T	<code>T</code>	τ	<code>\tau</code>
Θ	<code>\Theta</code>	θ	<code>\theta</code>	Υ	<code>\Upsilon</code>	υ	<code>\upsilon</code>
I	<code>I</code>	ι	<code>\iota</code>	Φ	<code>\Phi</code>	ϕ	<code>\phi</code>
K	<code>K</code>	κ	<code>\kappa</code>	X	<code>X</code>	χ	<code>\chi</code>
Λ	<code>\Lambda</code>	λ	<code>\lambda</code>	Ψ	<code>\Psi</code>	ψ	<code>\psi</code>
M	<code>M</code>	μ	<code>\mu</code>	Ω	<code>\Omega</code>	ω	<code>\omega</code>

一部のギリシャ文字の小文字には変体文字がある。

小文字	変体文字	コマンド	小文字	変体文字	コマンド
ϵ	ε	<code>\varepsilon</code>	ρ	ϱ	<code>\varrho</code>
θ	ϑ	<code>\vartheta</code>	σ	ς	<code>\varsigma</code>
π	ϖ	<code>\varpi</code>	ϕ	φ	<code>\varphi</code>

☆ その他 (記号 1)

以下の記号は、数式モード、数式立てで使用するここと。

出力	コマンド	出力	コマンド	出力	コマンド
$\leq$	<code>\leq</code>	$\because$	<code>\because</code>	$\leq$	<code>\leqq</code>
$\geq$	<code>\geq</code>	$\geq$	<code>\geq</code>	$\therefore$	<code>\therefore</code>
$\pm$	<code>\pm</code>	$\times$	<code>\times</code>	$\in$	<code>\in</code>
$\ni$	<code>\ni</code>	$\supseteq$	<code>\supseteq</code>	$\subseteq$	<code>\subseteq</code>
$\cap$	<code>\cap</code>	$\cup$	<code>\cup</code>	$\subset$	<code>\subset</code>
$\supset$	<code>\supset</code>	$\uparrow$	<code>\uparrow</code>	$\downarrow$	<code>\downarrow</code>
$\rightarrow$	<code>\rightarrow</code>	$\leftarrow$	<code>\leftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\Downarrow$	<code>\Downarrow</code>	$\Rightarrow$	<code>\Rightarrow</code>	$\Leftarrow$	<code>\Leftarrow</code>
$\rightarrow$	<code>\to</code>	$\mapsto$	<code>\mapsto</code>	$\neq$	<code>\neq</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\circ$	<code>\circ</code>	$\bullet$	<code>\bullet</code>
$\oplus$	<code>\oplus</code>	$\otimes$	<code>\otimes</code>	$\emptyset$	<code>\emptyset</code>
$\bot$	<code>\bot</code>	$\angle$	<code>\angle</code>	$\triangle$	<code>\triangle</code>

★ その他 (記号 2)

以下の記号は通常の入力で使用するものである (数式モード内で使用可否を含む)。

出力	コマンド	出力	コマンド	出力	コマンド	出力	コマンド
#	<code>\#</code>	\$	<code>\\$</code>	%	<code>\%</code>	&	<code>\&</code>
-	<code>_</code>	{	<code>\{</code>	}	<code>\}</code>	§	<code>\S</code>
£	<code>\pounds</code>	œ	<code>\oe</code>	ø	<code>\o</code>	Ø	<code>\O</code>
T _E X	<code>\TeX</code>	L _A T _E X	<code>\LaTeX</code>	L _A T _E X 2 _ε	<code>\LaTeXe</code>	©	<code>\o</code>

★ フォントサイズ

フォントのサイズは { } で囲み、`\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize` (標準), `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge` で指定する。例えば、`{\Large OUS}` とすれば OUS となる。それぞれのサイズは以下のようになる。

`\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`

★ 囲み

複数行の文字や数式などを囲む場合には、以下のコマンドを使う。このとき、[1] は “イチ” ではなく “エル” で、left を意味している。

タイトルを付ける場合

```
\begin{itembox}[1]{タイトルを付ける場合}
内容
\end{itembox}
```

または

```
\begin{screen}
内容
\end{screen}
```

のように、タイトルを付ける場合と付けない場合で分かれる。また、screen を shadebox もしくは boxnote に変更すると枠の形状が変わる。

shadebox にした場合

boxnote にした場合

3.3 図形と画像

3.3.1 T_EX 図形

TikZ

T_EX で図を描くには `picture` 環境や `METAPOST` などが使われてきた。この講義ではより高度なグラフを描くことができる TikZ(ていくす) を用いる。いろいろな例を見ながら説明する。ちなみに、TikZ のマニュアルは

<http://ftp.yz.yamagata-u.ac.jp/pub/CTAN/graphics/pgf/base/doc/>

に、`pgfmanual.pdf` というファイルで置いてあるが、1200 ページを超える量 (9.7MB) があり、目的のコマンドを探すのは難しいかもしれない。そこで、

<https://www.ctan.org/pkg/visualtikz>

にある `The document itself` を見るのが良いかもしれない。こちらは、コマンドと結果が表示されている。

☆ 簡単な図形を書く場合

簡単な直線や円を書く場合は、座標や半径が解れば、それらをもとに書くことができる。

例 3.3.1. (TikZ の例 1) 以下のソースファイルでは、 $(0,0)$, $(0,2)$, $(1,0.5)$, $(2,2)$, $(2,0)$ を結ぶ線分と、原点を中心に半径 1 の円を描いている。

TeX ソース (12)

```
%documentclass[dvipdfmx,a4j,12pt]{jarticle}
%usepackage{tikz}
%begin{document}
%tikz%draw (0,0)--(0,2)--(1,0.5)--(2,2)--(2,0);
%tikz%draw (0,0) circle (1);
%end{document}
```

コンパイル (12)



TikZ を用いて図を作成する場合、まずプリアンブルに

```
%usepackage{tikz}
```

を追加する (上の例の 2 行目)。そして、図を描きたい場所で コマンドを入力する。

この例ではまず、`\tikz\draw` のコマンドで、座標 $(0,0)$, $(0,2)$, $(1,0)$, $(2,2)$, $(2,0)$ を結ぶ線を描いている。次に、`\tikz\draw` のコマンドで、 $(0,0)$ を中心とする半径 1 の円を描いている。

$(0,0)$ の位置が 2 つの図で異なるのは、それぞれの図に異なる図形領域が確保されるからである。重ねて描く場合は、後述する ☆ 複数の図を 1 つの領域に描く場合 を利用する。

例 3.3.2. (TikZ の例 2)

`\draw` のあとに `[ ]` で囲み、オプションを指定できる。以下の例では角を 12pt 分丸くし、線の終わりを矢印 `->` にしている。

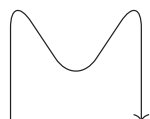
線の両端または一方を矢印にする方法は、`->` 以外に、`<-`, `<->`, `->>` などがある。

TeX ソース (13)

```
\documentclass[dvipdfmx,a4j,12pt]{jarticle}
\usepackage{tikz}
\begin{document}
\tikz\draw[rounded corners=12pt,->]
(0,0)--(0,2)--(1,0.5)--(2,2)--(2,0);
\end{document}
```

この実行結果は 以下の図が描かれる。

コンパイル (13)

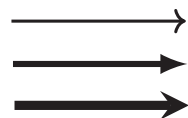


線の幅は `line width` で指定できる。また、矢印の種類も以下のようなものがある。

TeX ソース (14)

```
\documentclass[dvipdfmx,a4j,12pt]{jarticle}
\usepackage{tikz}
\begin{document}
\tikz\draw[line width=1pt,->] (0,0)--(2,0);
\tikz\draw[line width=2pt,-latex] (0,0)--(2,0);
\tikz\draw[line width=width=3pt,-stealth] (0,0)--(2,0);
\end{document}
```

コンパイル (14)



☆ 複数の図を1つの領域に描く場合

`\draw` コマンドなどを `\begin{tikzpicture}` と `\end{tikzpicture}` で囲む。その際に、例で挙げていた `\tikz\draw` の `\tikz` を省き、`\draw` だけをかく。

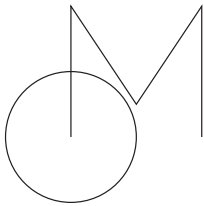
例 3.3.3. (TikZ の例 3) 例 3.3.1 をまとめて描くと次の例となる。

TeX ソース (15)

```
\documentclass[dvipdfmx,a4j,12pt]{jarticle}
\usepackage{tikz}
\begin{document}
\begin{tikzpicture}
\draw (0,0)--(0,2)--(1,0.5)--(2,2)--(2,0);
\draw (0,0) circle (1);
\end{tikzpicture}
\end{document}
```

この場合、2つの図の座標 (0,0) が一致していることが解る。

コンパイル (15)

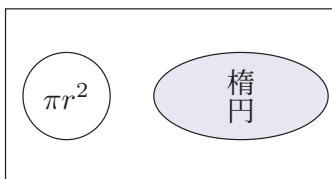


例 3.3.4. 以下は TikZ 部分のみと、それを含んだ実行結果を示している。

TeX ソース (16)

```
\begin{tikzpicture}
\draw(0,0) circle(0.5) node {$\pi r^2$};
\fill[blue!10](2,0) ellipse(1 and 0.5);
\draw(2,0) ellipse(1 and 0.5);
\draw(2,0) node {\hbox{\tate 楕円}};
\draw(-0.7,-1) rectangle(3.1,1);
\end{tikzpicture}
```

コンパイル (16)



TeX ソース (17)

```

\begin{tikzpicture}[x=1mm,y=1mm]
  \fill[gray] (0,0) circle (1)
    (10,10) circle (1)
    (20,10) circle (1)
    (20,0) circle (1);
  \draw(0,0)--(10,10)--(20,10)--(20,0);
  \draw[line width=2pt] (0,0)..controls (10,10) and (20,10)..(20,0);
  \draw[line width=2pt,gray] (0,0)..controls (10,10)..(20,0);
  \draw(10,-5) node {\hbox{ベジエ曲線}};
\end{tikzpicture}

```

コンパイル (17)



ベジエ曲線

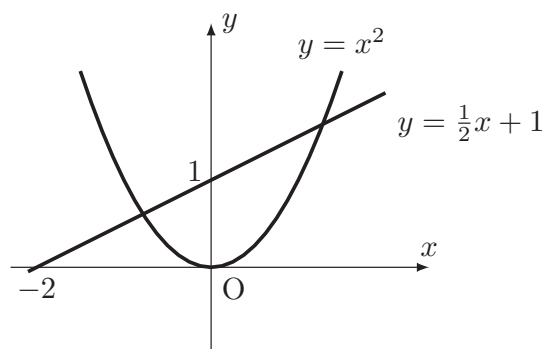
TeX ソース (18)

```

\begin{tikzpicture}
  \draw[-latex] (-2.3,0)--(2.5,0)node[above]{$x$};
  \draw[-latex] (0,-1)--(0,2.8)node[right]{$y$};
  \draw (0,0)node[below right]{0};
  \draw (-2,0)node[below]{$-2$};
  \draw (-0.4,1.1)node[right]{$1$};
  \draw[very thick,domain=-2.1:2]
    plot(\x,\x/2+1)node[below right]{$y=\frac{1}{2}x+1$};
  \draw[very thick,domain=-1.5:1.5]
    plot(\x,\x*\x)node[above]{$y=x^2$};
\end{tikzpicture}

```

コンパイル (18)



☆ 繰り返し

同じような線を沢山描きたい場合は、`\foreach` が便利である。

`\foreach \変数 in {a,b,...,c} object`

これは、変数を a から c まで、増分 $b-a$ で変化させ、`object` を実行させるコマンドとなる。

例 3.3.5. (`foreach` の使用例)

TeX ソース (19)

```
\begin{tikzpicture}[x=1mm,y=1mm]
\draw (0,0)--(100,0);
\foreach \x in {0,...,100} \draw(\x,0)--(\x,3);
\foreach \x in {0,5,...,100} \draw(\x,0)--(\x,5);
\foreach \x in {0,10,...,100} \draw(\x,0)--(\x,7);
\end{tikzpicture}
```

3 行目は変数 x を 0 から 100 まで増分 1 で変化させ、 $(x, 0)$ から $(x, 3)$ まで直線を描いている。

4 行目は変数 x を 0 から 100 まで増分 5 で変化させ、 $(x, 0)$ から $(x, 5)$ まで直線を描いている。

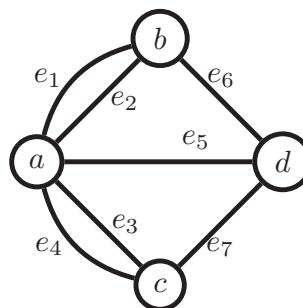
5 行目は変数 x を 0 から 100 まで増分 10 で変化させ、 $(x, 0)$ から $(x, 7)$ まで直線を描いている。

コンパイル (19)



例 3.3.6. 同様に、次のような図も描ける。

2		7						6
	9		4			8		
		5	3	6	1			
		9				4		1
5	2		8	1			3	7
1			4	8	7	6		
9	5	2		4	8	7		3
3	6		7			1	2	
8			6	2	3	5	9	4



3.3.2 画像ファイルの貼り付け

文中に画像を張り付けたい場合 (例えば  のように)、

`\includegraphics[オプション]{ファイル名}`

を使う。

[オプション] には

bb : 画像サイズの情報 , width : 表示幅 , height : 表示高さ
scale : 拡大縮小率 , angle : 回転角 , clip : はみ出しを切り取る

などがある。オプションを複数使う場合は、`,` で区切る。

bb の指定方法は、画面左下の座標、右上の座標で指定する。例えば、画像サイズが 640×480 の画像なら、

```
[bb=0 0 640 480]
```

とする。この講義では必ず指定すること。

表示幅, 表示高さは縦横の比率を変えて拡大や縮小したい場合や表示サイズを指定したい場合など に 使用する。例えば、画像サイズが 640×480 の画像を縦方向だけ半分にしたい場合は、

```
[bb=0 0 640 480, width=640pt,height=240pt]
```

とする。縦横の比を保って拡大 (縮小) したい場合は scale を使う。例えば 30% にしたい場合は、

```
[bb=0 0 640 480, scale=0.3]
```

とする。

画像を回転させたい場合は angle (と origin) を使う。angle = n で回転角 n を指定し (反時計方向回りを +)、回転の中心は origin = oz で指定する。oz は図の中心 [c], 左上 [tl], 右上 [tr], 左下 [bl], 右下 [br] のいずれかの ([] 内の) 値を入れる。[デフォルトは bl]

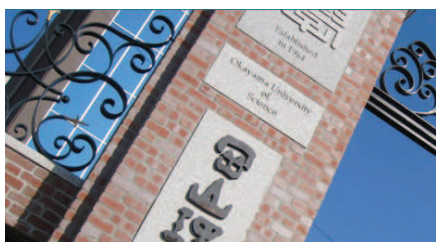
ファイル名の場所には、ファイル形式が EPS, BMP, JPEG などの画像のファイル名を書く。



```
\includegraphics[bb=0 0 228 236, angle=30,origin=c]{tanQ.bmp}
```

また、レポートや論文などで画像の位置を自動的に調節して表示させる場合は、画像を貼り付けたい 辺りに

```
\begin{figure}[位置指定]
```



```

\includegraphics[オプション]{ファイル名}
\end{figure}

```

と入力する。“辺り”と言ったのは、画像の貼り付け場所を [位置指定] で以下の4つから指定するためである。複数書く場合、書く順は関係なく、優先順位は h,t,b,p の順である。

h : 記述した場所に表を出力 , t : ページの上端に表を出力

b : ページの下端に表を出力 , p : 表専用のページを用意して出力

例 3.3.7. たとえば、画像サイズが 640x480 の画像を 30% に縮小し、そのページの上端に張り付ける場合のコマンドが以下となる。

```

\begin{figure}[t]
\includegraphics[bb=0 0 640 480,scale=0.3]{ous.bmp}
\end{figure}

```

実際にはこの場所にこの (↑) コマンドを書いているが、表示されるのはこのページの上端になっている。

また貼り付けるための画像として、TikZ にはいろいろな素材が用意されている。例えば…。



```

\begin{tikzpicture}[gray]
\node[name=a,shape=bob,minimum size=1.5cm] {};
\node[ellipse callout, draw, callout absolute pointer={(a.mouth)}]
  at (3,.5) {我々も TikZ だよ。};
\node[name=b,alice,minimum size=1.5cm,mirrored] at (8,0) {};
\node[ellipse callout, draw, callout absolute pointer={(b.mouth)}]
  at (5.8,-.0.3) {すごいね!};
\end{tikzpicture}

```

3.4 プレゼンテーション

3.4.1 Beamer とは

$\text{\TeX}$ でプレゼンテーションを行う場合、以前は slides や seminar が使われていた。

最近では、powerdot や Beamer などのクラスがよく使われている。ここでは、Beamer を使用した方法の説明を行う。まずは例を述べ、そのあと個別に説明を行う。

例 3.4.1. (Beamer の使用例)

説明のために、行頭に行番号を付けているが、実際は入力しない。

```

1 %documentclass[dvipdfmx,cjk]{beamer}
2 %usetheme{Antibes}
3 %title{Beamer によるプレゼンテーション}
4 %author{応用 数学}
5 %institute[岡山理科大学]
6
7 %begin{document}
8 %begin{frame}{}
9 %titlepage
10 %end{frame}
11 %begin{frame}{まずは}
12 最初のページ。
13 %end{frame}
14 %begin{frame}[t]{その次}
15 2 ページ目
16 %end{frame}
17 %end{document}

```

まず、1 行目はいつもの呪文。

2 行目はメイン・テーマの指定。省略可能で、省略すると default となる。

プリアンプルには メイン・テーマの他にカラー、フォント、インナー、アウターのテーマも指定できる。

3,4,5 行目は発表タイトル、著者名、所属等である。

6 行目 (`%begin{document}` の前) に空行が無い場合、エラーとなる。

8~10 行がタイトル、11~13 行が 1 ページ目、14~16 行が 2 ページ目となる。

11 行目や 14 行目の `{frame}` の後の `{ }` 内は各ページのタイトルを示す。また、14 行目の `{frame}` の後の `[t]` は、フレーム内のどの位置 `[top, center, bottom]` (上詰め, 中央, 下詰め) に表示するかを指定している。省略すると `[c]` となる。



3.4.2 Beamer のコマンド

★ frame

Beamer では、1つの画面を frame と呼び、`\begin{frame}` と `\end{frame}` の区切りで 1つの frame を作る。(基本的に) この frame 内を 1 ページとして表示される。frame 環境の中では、通常の L^AT_EX のコマンドを使用する。例えば、以下のような frame を考える。

```
\begin{frame}[options]{はじめに}
```

ここには、通常の L^AT_EX のコマンドを使用することができる。

.....

```
\end{frame}
```

この options にはさまざまなオプションが用意されているが、t,c,b を良く使う。

★ ブロック

L^AT_EX で普通に使用していた囲みも使えるが、Beamer ではブロックが使える。

基本的なブロック囲みは `\begin{block}` と `\end{block}` で囲む。

例 3.4.2. (block の例)

```
\begin{block}{タイトル}
```

基本的なブロック

```
\end{block}
```

タイトル

基本的なブロック

また、`\begin{block}`, `\end{block}` の block を `exampleblock`, `alertblock` に変えると以下ようになる。

◎ `\begin{exampleblock}{ }` ~ `\end{exampleblock}`

exampleblock

例えば、例につかうブロック

◎ `\begin{alertblock}{ }` ~ `\end{alertblock}`

alertblock

例えば、注意につかうブロック

★ 一時停止

1つのフレーム内で、表示を一時停止する場合に `\pause` を使う。スペースキーを押すことによって、`\pause` 以下を表示する。

(一時停止の例)

```
\begin{frame}{一時停止}
  まず、この部分は表示される。
  \pause
  この部分は、スペースキーが押されるまで表示されない。
\end{frame}
```

★ オーバーレイ

`\pause` よりも表示の順番を細かく設定する方法として、`itemize` を使用する。次の例に示すように、表示させたいフレーム番号を書く形で指定する。

例 3.4.3. (オーバーレイの例. 説明のため行番号付き)

```
1 \begin{itemize}
2   \item<1-> その1
3   \item<2-3> その2
4   \item<3> その3
5   \item<4> 2,3を消して、その4
6   \item<1,4> その1とその4のとき表示
7 \end{itemize}
```

1つのフレームの中に、`\begin{itemize}` と `\end{itemize}` を入れると、このフレームの中にサブフレームが出来るイメージになる。この例では4枚のサブフレームが出来ていて、

2行目の `\item<1->` に続く“その1”は1フレーム目以降、表示される。

3行目の `\item<2-3>` に続く文字は2フレーム目と3フレーム目に表示される。

4行目の `\item<3>` に続く文字は3フレーム目のみ表示される。

5行目の `\item<4>` に続く文字は4フレーム目のみ表示される。

6行目の `\item<1,4>` に続く文字は1フレーム目と4フレーム目に表示される。

★ 色付け

L^AT_EX と同様に文字に色付けをすることが出来るが、Beamer では標準で red, blue, green, cyan, magenta, yellow, black, darkgray, gray, lightgray, orange, violet などがある。

使用方法は `\textcolor{上記の色}{色付けしたい文字}` であり、例えば、赤色の文字は

```
\textcolor{red}{赤色の文字}
```

とする。

3.4.3 二段組

ページを分割し、二段組にすることが出来る。試験問題などで使われることがある。

二段組をする場合は、プリアンブルに `\usepackage{multicol}` を書き、二段組をしたい場所に `\begin{multicols}{2}`^{*3} と `\end{multicols}` を入れる。その中に、まず左側の内容を入力し、左側が終わったら `\columnbreak` を書き、右側に移る。

変換後の pdf ファイルでは、真ん中に線が引かれ左右に分けて書かれる。

例 3.4.4. (二段組みの例)

以下の問いに答えよ。

```
\begin{multicols}{2}
  \textbf{問題 1}. 64MiB の容量を....
  ....
\columnbreak
  \textbf{問題 2}. 次の数式を....
  ....
\end{multicols}
```

表現とメディアの数理 定期試験

学生番号 _____ 氏名 _____

以下の問いに答えよ。

問題 1. 64MiB の容量をもつ USB メモリに、日本語が何文字保存できるか答えよ。
ただし、メディアフォーマット等に必要な領域は 0 とする。

問題 2. 次の数式を表示させる L^AT_EX コマンドを答えよ。

$$(1) \sum_{i=1}^{10} i = 55$$

$$(2) \int a^x dx = \frac{a^x}{\log a} + C$$

この命令文の前に次のコマンドによって、左右の間隔や区切り線の太さを指定できる。

`\setlength{\columnsep}{10.0pt}` で左右コラムの間隔の指定

`\setlength{\columnseprule}{0.4pt}` で区切り線の太さの指定

♠ 補足 なお、1 ページ全体や 1 つのソース全体を二段組にする場合は、`\twocolumn` ～ `\onecolumn` を用いたり、`twocolumn` クラスを使う。

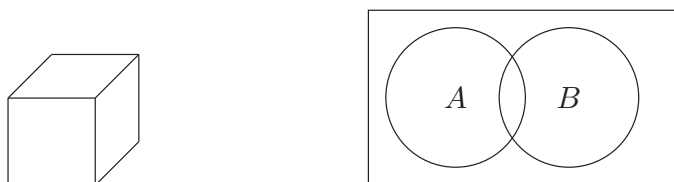
^{*3} この 2 を 3 にすると三段組みになる。

3.5 L^AT_EX 課題

課題 3.5.1. A4 用紙に次項にある課題の枠囲みの中が出力されるように T_EX ソースファイルを作成せよ。ただし、最後の S00M000 氏名 は各自の学生番号, 氏名に変えること。

また、文章の改行の位置が異なるかもしれないが、そのまま構わない。

課題 3.5.2. 以下の図が出力となるように T_EX ソースファイルを作成せよ。ただし、横に並べる必要はなく、大きさや線の太さもそれぞれ異なって構わない。



課題 3.5.3. 好きな図を考え、その図が出力となるように課題 3.5.2 の T_EX ソースファイルの続きに作成せよ。

その際、作成した 3 つの図が 1 枚の A4 用紙にまとまるようにすること。

課題 3.5.4. (1) 例 3.4.1 を参考に、プレゼンテーション用の L^AT_EX ソースファイルを作成せよ。

1 ページ目はタイトルがくること。

2 ページ目には大学生活や卒業研究、教職についてなど何かコメントを上詰めで書くこと。

3 ページ目には itemize を用いて、右のように表示される

1, 2, 3, 4, 5, 6, 7 に対して、1 回目は 1 のみ、2 回目は偶数のみ、3 回目は奇数のみ、4 回目は素数のみ、5 回目は全てが表示されるページを作ること。

4 ページ目にはブロックを用いて何か定義か定理を挿入すること。

課題 3.4.4 の 3 ページ目の説明

- 1
- 2
- 3
- 4
- 5
- 6
- 7

(2) (1) で作成したものに対して、メイン・テーマ (例 3.4.1 の 2 行目) をネットで調べ、Antibes 以外のものに変更せよ。

このとき、メイン・テーマは、日本語環境では使えないものもある。いくつか試し、正しく表示されるものを選ぶこと。

また、`\author{応用 数学}` の “応用 数学” は各自の学生番号, 氏名に変更せよ。

課題

課題プリント

三角関数の加法定理

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta \quad (1)$$

$$\sin(\alpha - \beta) = \sin \alpha \cos \beta - \cos \alpha \sin \beta \quad (2)$$

$$\cos(\alpha + \beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta \quad (3)$$

$$\cos(\alpha - \beta) = \cos \alpha \cos \beta + \sin \alpha \sin \beta \quad (4)$$

定義 1.1 f を开区間 I 上の関数とする。 f が $x_0 \in I$ で 微分可能であるとは

$$\lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}$$

が有限値として存在するときをいう。

定理 1.1

$\{f_n\}_{n=1}^{\infty}$ を $[a, b]$ 上の関数列で、 $[a, b]$ 上の関数に一様収束しており、各 f_n は有界可積分関数であるとする。このとき、極限関数も有界可積分であって

$$\lim_{n \rightarrow \infty} \int_a^b f_n(x) dx = \int_a^b \lim_{n \rightarrow \infty} f_n(x) dx$$

が成り立つ。とくにベキ級数 $\sum_{n=0}^{\infty} a_n(x - x_0)^n$ に対しては、その収束区間において

$$\int \sum_{n=0}^{\infty} a_n(x - x_0)^n dx = \sum_{n=0}^{\infty} \frac{a_n}{n+1} (x - x_0)^{n+1} + C$$

が成立する。

定義 1.2 (正則変換、正則行列)

1 次変換 f

$$f : \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

において、 $ad - bc \neq 0$ のとき、 f は逆変換を持ち、その逆変換 f^{-1} は

$$f^{-1} : \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

となる。このように表現行列に逆行列が存在する 1 次変換を正則変換といい、正則変換の表現行列を正則行列という。

以上の作成を L^AT_EX 2_εで行いました。S00M000 氏 名