

L^AT_EXによる文書作成入門

山形大学工学部情報・エレクトロニクス学科
情報・知能コース

目次

1 T_EX とは	3
1.1 目的	3
1.2 T _E X の種類	3
1.3 L ^A T _E X の特徴	3
1.4 L ^A T _E X の処理過程	4
2 L^AT_EX の基礎	4
2.1 ソースファイルのサンプル	4
2.2 使ってみる	5
3 L^AT_EX のコマンド	7
3.1 プリアンブル	7
3.2 文字、文、文章、段落	8
3.3 節、小節	8
3.4 タイトル、目次	9
3.5 打ち込んだとおりに出力する	10
3.6 箇条書き	10
3.7 中央寄せとスペーシング	11
3.8 表組み	12
4 実習操作 その1	13
5 数式	14
5.1 数式環境	14
5.2 数式のコマンド	15
5.3 数式の参照	16
5.4 場合分け、行列	17

6 外部環境の利用	18
6.1 usepackage	18
6.2 図	19
6.2.1 tgif を用いて図を描く	19
6.2.2 図を取り込む	20
7 参考文献	20
8 実習操作 その 2	21
9 レポート課題	22

1 $T_E X$ とは

1.1 目的

書籍や新聞をつくる行程で、職人が様々な活字を組んで印刷する版をつくる。この過程を組版という。 $T_E X$ はテキストファイルに埋め込まれたコマンドを解釈して組版を行う。通常のワープロソフト等では文章を純粋なテキストファイルではなく、それぞれの独自のバイナリ形式で保存していることが多い。それに対し、テキストファイルで保存されたデータは可搬性に優れており、他の整形ソフト等での加工が容易であるという側面を持つ。本実習では、 $T_E X$ の利用法を学習し、簡単なレポート程度の組版を $T_E X$ を使ってできるようになるのが目的である。

1.2 $T_E X$ の種類

$T_E X$ に後述する環境等のマクロセットをつけくわえたものが $LAT_E X$ である。本演習では、日本語 $LAT_E X$ のひとつである $pLAT_E X 2_E$ を扱うが、以降特に区別する必要が無い場合、単に $LAT_E X$ という。本テキストも $LAT_E X$ を利用し作成されている。

1.3 $LAT_E X$ の特徴

$LAT_E X$ の特徴(利点/欠点)を列挙してみると次のようになる。

- 組版が綺麗。特に数式を非常に美しく記述できる。
- 表、図、数式等の相互参照が非常に容易である。
- 表、図は別のソフト(*gnuplot*などの画像作成ソフト)で作成し、文章へ挿入するのが普通である。
- 目次、脚注の作成機能が優れている。
- ソースファイルは(テキストファイルなので)任意のエディタで作成可。
- マシンパワーを必要とせず、必要なストレッジも少なくすむ。
- WYSIWYG でない。
- 「コマンド」が多く、ソースファイルが繁雑になりやすい。
- コンパイルする必要がある。

WYSIWYG とは “What You See Is What You Get” の略で、見た目を確認しながら編集する方式のことをいう。MS Word や一太郎といったワープロソフトはこの編集方式である。

エディタとは、文書作成プログラムのことを指すが、通常テキストファイル(ASCII コード, EUC, UTF-8 コードなどの文字コードデータのみで構成されるファイルを作成できるものを言う。

1.4 *L^AT_EX* の処理過程

L^AT_EX による原稿の入力から出力までの過程は次のようになる。

原稿の入力 エディタにより、地の文章、および数式出力や文字修飾、整形といった *T_EX* コマンドを入力し、拡張子 *.tex* のファイルとして保存する。これを *T_EX* ソースファイルと呼ぶ。

タイプセット 入力されたコマンドをもとに組版を行う。*T_EX* の場合は、*T_EX* でコンパイルする、*T_EX* にかける、とも言う。コンパイルの途中でエラーが出た場合は、当該ソースファイルを修正する。これを繰り返しコンパイルに成功したら、DVI 形式のファイルができる。これは拡張子 *.dvi* をもつファイルで、DeVice Independent の略。単に *dvi* ファイルと呼ぶことが多い。

プレビュー できあがった *dvi* ファイルの印刷イメージをプレビューアにより確認する。通常プレビューアとしては *xpdf* を用いる。

出力 DVI 形式のファイルを印刷するためには、Postscript 言語という業界標準のページ記述言語によって書かれたファイル形式(単に *ps* ファイルと呼ぶ)に変換する必要がある。この演習では *dvips* を用いて *ps* ファイルに変換する。その後、プリンタドライバにより、プリンタにデータを送信し、印刷を行う。

2 *L^AT_EX* の基礎

2.1 ソースファイルのサンプル

全ての英数字および、“*.,;:\{\}%*”などの記号は ASCII コード(つまり、通常半角文字と呼ばれるもの)で入力する。ソースコード内の改行は基本的に無視されるので、適宜挿入してソースをみやすくする。また % で始められた行は、コメント行で、文末まで無視される。半角スペースは複数個入力しても一個のスペースとして解釈される。*T_EX* のコマンドは

バックスラッシュ + コマンド名

の形をしている。引数をとるときは、*\command{arg}* のように、引数 *arg* を指定する。*[opt]* の形のオプション引数をとることもある。コマンド名の部分には、大小英文字のみで、数字や特殊文字を含めることはできない。特に *\begin{環境名}* と *\end{環境名}* は環境と呼ばれる。

表 2-1: hello.tex

1: \documentclass[12pt,a4j]{jarticle}	L ^A T _E X の基本形
2: \title{LaTeX の基本形}	
3: \author{a00919 山形 太郎}	a00919 山形 太郎
4: \date{2006 年 3 月 31 日}	2006 年 3 月 31 日
5:	
6: \begin{document}	
7: \maketitle	
8: ここに内容を書きます。	ここに内容を書きます。
9: \end{document}	

2.2 使ってみる

表 2-1 のソースファイルを “hello.tex” というファイル名で保存し、(1) コンパイル、(2) プレビュー、(3) 印刷をこの順に行ってみよう。以下に入力例を示す。

eib-pc001:~> emacs hello.tex &	← エディタでファイルを編集
eib-pc001:~> plateX hello[.tex]	← コンパイル
eib-pc001:~> xdvi hello[.dvi] &	← プレビュー
eib-pc001:~> dvips hello[.dvi]	← ポストスクリプトに変換
eib-pc001:~> lpr hello.ps	← 印刷

ここで、hello[.tex] は hello もしくは hello.tex のどちらでも良いことを意味している。

pdf ファイルを作成したいときは dvipdfmx を使う。

eib-pc001:~> dvipdfmx hello[.dvi]	← PDF に変換
eib-pc001:~> acroread hello.pdf &	← プレビュー

実習 2.1 hello.tex の(1) タイトルを「実習 2.1」、(2) 著者を自分の学籍番号+名前、(3) 日付を今日、にそれぞれ修正して TeX ソースを作成し、コンパイル、印刷をしなさい。

エラーが起きた場合

コンパイルを実行すると画面に(英語で)コンパイルメッセージが流れる。ソースファイルに間違いがあった場合には、エラーメッセージが表示され、プロンプト?があらわれ、停止する。通常は x を入力して、処理を中断し、表示されたエラーをエディタで修正し、再度コンパイルを試みる。これを、コンパイルが成功するまで繰り返す。

```
eib-pc001:~> cat errsam.tex
\documentclass{jarticle}
```

```
\begin{document}
\begin{center}
This is a center.
\end{center}
\end{document}

eib-pc001:~> plateX errsam
pLaTeX2e <2017/10/28>+4 (based on LaTeX2e <2017-04-15>)
Babel <3.18> and hyphenation patterns for 84 language(s) loaded.
(/usr/share/texlive/texmf-dist/tex/plateX/base/jarticle.cls
Document Class: jarticle 2017/09/19 v1.7g Standard pLaTeX class
(/usr/share/texlive/texmf-dist/tex/plateX/base/jsize10.clo) (./errsam.aux))
Runaway argument?
{center This is a center. \end {center} \end {document}
! File ended while scanning use of \begin.
<inserted text>
          \par
<*> errsam

? x
No pages of output.
Transcript written on errsam.log.
eib-pc001:~>
```

← "x" と入力して ENTER を押す

次のように “**” というプロンプトが出て停止した場合、Ctrl+c を入力して上のエラー画面に戻す。

```
This is e-pTeX, Version 3.14159265-p3.7.1-161114-2.6 (utf8.euc) (TeX Live 2017/Debian) (pre)
restricted \write18 enabled.
entering extended mode
./hoge.tex
pLaTeX2e <2017/10/28>+4 (based on LaTeX2e <2017-04-15>)
Babel <3.18> and hyphenation patterns for 84 language(s) loaded.
)
*^C
! Interruption.
<*>

? x
No pages of output.
Transcript written on hoge.log.
```

← "Control"を押しながら"c" を入力して ENTER を押す。

組版処理に成功すると、ソースファイルがあったディレクトリに、次のファイルが作成される。

拡張子が .aux: 補助ファイル (Auxiliary)。相互参照用の図表番号などの管理情報が書きこまれる。

拡張子が .log: 記録ファイル (Log)。TeX のコンパイル中表示されているメッセージを保存している。エラーの詳細を知りたいときに参照する。

表 2-2: `\documentclass` のオプション

文字の大きさ	<code>10pt,11pt,12pt</code> など。
用紙サイズ	<code>a4paper,a5paper,b4paper,a4j,a5j,b4j</code> など。
二段組	<code>twocolumn</code> など。

拡張子が`.dvi`: DVI 形式のファイル。

正常にコンパイルできたなら、補助ファイル(`.aux`)、記録ファイル(`.log`)は削除してかまわない。

3 *LATEX* のコマンド

理系の論文やレポート作成をする際に繰り返し必要となるコマンドを紹介する。

3.1 プリアンブル

表 2-1 のソースのように、*LATEX* のソースは先頭に

```
\documentclass[オプション]{クラス}
```

が必要である。クラスは文章の種類を設定するところで、通常 `jarticle` を指定する(他に、`book`, `report` 等が指定できる)。オプションは全体の組み方の指定をする。主に表 2-2 のようなものが指定できるが、通常は`\documentclass[12pt,a4j]{jarticle}` 程度で十分である。

次に、本文全体は`\begin{document}` と`\end{document}` で囲まれている必要がある。結局、*LATEX* のソースファイルは次のような格好をしている:

```
\documentclass[12pt,a4j]{jarticle}
(A)
\begin{document}
(B)
\end{document}
```

これらは、すべての *LATEX* のソースで共通なので、今後の説明では (B) のところのみ記述している。したがってコンパイルの際はこれらを補わなければならないことに注意。なお、(A) の領域をプリアンブルといい、文章全体の細かな指定などを行う部分である。

表 3-3: 特殊文字の表示コマンド

特殊文字	そのまま使ったとき	表示コマンド
#	エラー	\#
\$	以降が数式環境となる	\\$, \textdollar
%	以降がコメント行となる	\%
&	エラー	\&
-	エラー	_
{	無視されるかエラー	\{
}	無視されるかエラー	\}
\	次の単語がコマンドとみなされる	\textbackslash
^	エラー	\textasciicircum
~	改行抑制空白	\textasciitilde
<	i	\textless
>	i	\textgreater
	—	\textbar

3.2 文字、文、文章、段落

以下の特殊文字を除き、基本的には入力文字は、それ自身を表示する。数式環境内とそれ以外では特殊文字の扱いが異なる(§5)。通常の環境内では、

#, \$, %, &, _, {, }, \, , ^, ~, <, >, |

は、特別な意味をもち、そのまま表示されないことに注意しよう。表示することが必要な場合には、それぞれ表示用のコマンド(表 3-3)を使うか、安直に後述の `verbatim` 環境を使う。英語の二重引用符は“‘と’”のように、引用符を二つ続けて書く。

文中には、表示のための文字列とコマンドが混在する。その際、コマンドと文字列がつながってエラーになることがあるので注意する(表 3-4)。

ソースファイル内では、改行は空白と同じかもしくは無視される。一行以上空けると改段落する。行の分割は `TEX` が自動的に判断する。どうしても手動で行を分割したい場合、その場所に `\par` または `\\"` を挿入する。

文頭または文中に %があらわれると `TEX` はそれ以降行末までを無視する。つまり、%はコメント行や、一時的にソースから削除したい部分を指定するのに使える。

3.3 節、小節

論文、レポートなど、文章全体で何かを説明する目的があるとき、文章を小テーマごとに節に分割し、小見出しをつけるのがよい。`article` スタイルでは、

表 3-4: コマンドと空白

	入力例	出力
正	\LaTeX_nician	L ^A T _E X_nician
誤	\LaTeXnician	(エラーになる)
正	\LaTeX{}nician	L ^A T _E Xnician
参考	\LaTeX{}_nician	L ^A T _E X nician

\section{}, \subsection{}, \subsubsection{}

が用意されている。それについて通し番号が付けられ、文字の大きさや行間の空きなどが自動で調整される。

— (section{} の出力例) —

```

1: \section{C プログラムの作り方}
2: C 言語は UNIX と非常に関係が深く、
3: ....
4: 以下 C プログラムの作り方を説明する。
5: \subsection{ソースの作り方}
6: エディタを用いてソースファイルを作る。
7: 通常、...
8: \subsection{コンパイルの仕方}
9: C のコンパイラは \verb|cc| を使う。
10: オプションとして...
11: \section{C プログラムのサンプル}
12: ...

```

1 C プログラムの作り方

C 言語は UNIX と非常に関係が深く、.... 以下 C プログラムの作り方を説明する。

1.1 ソースの作り方

エディタを用いてソースファイルを作る。通常、...

1.2 コンパイルの仕方

C のコンパイラは cc を使う。オプションとして...

2 C プログラムのサンプル

...

3.4 タイトル、目次

タイトル、日付および著者名は、プリアンブルに記述する。それぞれ、\title{}, \date{} および\author{} というコマンドが用意されている。これらを実際に表示させるコマンドは \maketitle で、\begin{document} の次に書くことが多い。

目次は、表示したい場所に\tableofcontents を書く。通常は、\maketitle の後くらいに配置する。その他、表目次、図目次が、それぞれ\listoftables, \listoffigures で表示される。これらを使った場合、補助ファイル(aux) に参照情報を書きこむ必要があるため、二回以上コンパイルする必要がある。

3.5 打ち込んだとおりに出力する

特殊文字を表示したい、C プログラムを引用したい、など打ち込んだとおりに出力する必要がある場合が多々ある。その場合、L^AT_EX では `verbatim` 環境を利用する。一行以上をまるごと表示する場合には、`\begin{verbatim}... \end{verbatim}` を、文中の一部だけを表示する場合には、`\verb` をそれぞれ利用する。`\verb` は使い方がちょっと特殊で、次に続く一文字が区切り文字になる。そこから同じ区切り文字がもう一度現れるまでが`\verb` の影響範囲内になる。

```

1: 次は C のサンプルである。
2: \begin{verbatim}
3: #include <stdio.h>
4: int main()
5: {
6:     int i=2;
7:     printf("%d %d\n", i, i*i) ;
8:
9:     return 0;
10: }
11: \end{verbatim}
12: ここで、\verb|#include| はプリプロ
13: セッサへの命令で… また、\verb+%d+
14: で整数型の変数を表す。

```

— (verbatim 環境の出力例) —

次は C のサンプルである。

```

#include <stdio.h>
int main()
{
    int i=2;
    printf("%d %d\n", i, i*i) ;

    return 0;
}

ここで、#include はプリプロセッサへの命令で… また、%d で整数型の変数を表す。

```

3.6 箇条書き

L^AT_EX では、記号付 (`itemize`)、番号付 (`enumerate`)、見出し付 (`description`) のそれぞれ箇条書き環境が用意されている。これらの環境内で、箇条書きの項目は次の例のように `\item[...]` で指定する。

```

1: 上杉家の名君として次の二人が有名。
2: \begin{description}
3: \item[上杉謙信] 1530 -- 1578 没。
4: 米沢上杉家の藩祖。名将と謳われる。
5: \item[上杉鷹山] 1751 -- 1822 没。
6: 第 9 代米沢藩主。中興の名君。
7: \end{description}

```

— (description 環境の出力例) —

上杉家の名君として次の二人が有名。

上杉謙信 1530 – 1578 没。米沢上杉家の藩祖。名将と謳われる。

上杉鷹山 1751 – 1822 没。第 9 代米沢藩主。中興の名君。

```

1: The latest CD by “PUR” consists
2: of 4 songs with the title:
3: \begin{enumerate}
4: \item Ouvert\“ure
5: \item Hold me tight
6: \item If you hear this Tango
7: \item Adventure land.
8: \end{enumerate}

```

— (enumerate 環境の出力例) —
The latest CD by “PUR” consists of 4 songs with
the title:

1. Ouvertüre
2. Hold me tight
3. If you hear this Tango
4. Adventure land.

```

1: 山形大学米沢キャンパス組織図：
2: \begin{itemize}
3: \item 工学部 (6 学科)
4: \begin{itemize}
5: \item 情報・エレクトロニクス学科
6: \item システム創成工学科
7: \item (他 4 学科略)
8: \end{itemize}
9: \item 大学院理工学研究科 (6 専攻)
10: \begin{itemize}
11: \item 数理情報システム専攻
12: \item 情報・エレクトロニクス専攻
13: \item (他 4 専攻略)
14: \end{itemize}
15: \end{itemize}

```

— (itemize 環境の出力例) —
山形大学米沢キャンパス組織図：

- 工学部 (6 学科)
 - 情報・エレクトロニクス学科
 - システム創成工学科
 - (他 4 学科略)
- 大学院理工学研究科 (6 専攻)
 - 数理情報システム専攻
 - 情報・エレクトロニクス専攻
 - (他 4 専攻略)

3.7 中央寄せとスペーシング

強調のため、ある単語または文を中心に置くには `center` 環境を用いる。

```

1: 強調のため、ある単語または文を
2: 中心に置くには
3: \begin{center}
4: center 環境
5: \end{center}
6: を用いる。

```

— (center 環境の出力例) —
強調のため、ある単語または文を中心に置くには
center 環境
を用いる。

T_EX では段落間や環境間のバランスも自動的にとられる。空白文字はいくつ入力しても
一つ分の空白しか出力しないが、どうしてもある一定の幅の空白を出力したい場合、い
くつかコマンドが用意されている(3-5)。

表 3-5: いろいろなスペーシング

\hspace{length}	横方向の長さ <i>length</i> の空白
\vspace{length}	縦方向の長さ <i>length</i> の空白
\hfill	横方向に入るだけ空白をいれる。
\vfill	縦方向に入るだけ空白をいれる。
\quad	横方向に“M”の幅くらいの空白をいれる。
\medskip	縦方向に平行くらいの空白をいれる。
\newpage	改ページする。

3.8 表組み

表を組むには tabular 環境を用いる。tabular 環境は、

```
\begin{tabular}{書式}
欄 1 & 欄 2 & 欄 3 & ... & 欄 n \\
欄 1 & 欄 2 & 欄 3 & ... & 欄 n \\
...
欄 1 & 欄 2 & 欄 3 & ... & 欄 n
\end{tabular}
```

の文法をもつ。書式は、c(中央寄せ), l(左寄せ), r(右寄せ), |(縦線) をくみあわせてつくる。*clr* を指定した数だけ欄がつくられる。各行では、欄を入力したあと、& で区切る。表における一行分の欄の最後に \\ を指定する。\\hline で横線を引くことができる。

```
1: \begin{center}
2: \begin{tabular}{r|lc}
3: \hline
4: 国名 & 首都 & 通貨 \\
5: \hline
6: 日本 & 東京 & 円 \\
7: イギリス & ロンドン & ポンド \\
8: アルゼンチン&ブエノスアイレス&ペソ\\
9: \hline
10: \end{tabular}
11: \end{center}
```

— (tabular 環境の出力例) —

国名	首都	通貨
日本	東京	円
イギリス	ロンドン	ポンド
アルゼンチン	ブエノスアイレス	ペソ

表にキャプション (=簡単な表の説明) や表番号をつけたい場合は、table 環境を使う。table 環境はフロートである。つまり印刷位置は固定でなく、コンパイル時に決定される。環境内ではキャプションは\caption{説明}を指定することにより表示される。

```

1: \begin{table}
2: \caption{各国の首都と通貨}
3: \begin{center}
4: \begin{tabular}{r|lc}
5: \hline
6: 国名 & 首都 & 通貨 \\
7: \hline
8: 日本 & 東京 & 円 \\
9: イギリス & ロンドン & ポンド \\
10: アルゼンチン&ブエノスアイレス&ペソ\\
11: \hline
12: \end{tabular}
13: \end{center}
14: \end{table}

```

表 1: 各国の首都と通貨

国名	首都	通貨
日本	東京	円
イギリス	ロンドン	ポンド
アルゼンチン	ブエノスアイレス	ペソ

表番号の参照は後述するように、数式番号、図番号の参照と同様にできる(§5.3).

4 実習操作 その1

実習 4.1 次の文章をできるだけ忠実に再現するように、 \TeX のソースを作成し、コンパイルとディスプレイへの表示を実行せよ。

109 キーボードの第一列には数字が並んでいるが、 shift キーを押しながらでは、順に、

! “ # \$ % & ’ () = ~ |

が入力される。(ちなみにこれは verbatim 環境を利用すると、

! “ # \$ % & ’ () = ~ |

という表示になる。) これが、101 キーボードでは、

! @ # \$ % ^ & * () - + | ~

と異なる順に配置されている。

実習 4.2 第 3.5 節から第 3.8 節までにある枠囲みの \TeX ソースを入力し、コンパイルとディスプレイへの表示を実行せよ。その際、 \TeX ソースは \subsubsection {小節} を利用して各小節に分けなさい。小節には、テキストのその節自身のタイトルまたは使用した環境名など内容を表しているタイトルを用いること。

実習 4.3 `\textit{...}`, `\textbf{...}` で文もしくは単語をくくるとそれぞれ、斜字体、ボールド体になる。

- (1) 次の文は英英辞典のある一部である。この T_EX ソースを作成し、コンパイルとディスプレイへの表示を実行せよ。

cat (*pl. cats*)

1. A **cat** is a small, furry animal with a tail, whiskers, and sharp claws. Cats are often kept as pets.
2. **Cats** are lions, tigers, and other wild animals in the same family.
3. *See also Cheshire cat, fatcat, wildcat.*

- (2) `verbatim` 環境を使い、(1)で作成したソースをそのままディスプレイへ表示させなさい。ただし、ソースは表示枠からはみ出さないように調整すること。

5 数式

作者の Knuth が数式を表示する目的で、T_EX を開発しただけあって、綺麗な数式の表示は他の追随を許さないものがある。数式環境内では、通常の環境内と違って特殊文字の扱いが異なり、`<,>` はそのまま不等号を表し、`|` は縦棒を表す。半角スペースは表示されず、コマンドを区切ったり、見易くするために使われる。改行は自由だが、数式環境内には空行があつてはならない。

5.1 数式環境

一行立ての数式は、`\[` と `\]`、または `\begin{equation}` と `\end{equation}` を使う。文中の一部だけ数式にしたい場合は、その部分を `$...$` で囲む。複数行の数式を書きたい場合は `eqnarray` 環境、`eqnarray*` 環境を使う。この環境内では改行を `\\"` で明示的に指定する必要がある。`equation` 環境および `eqnarray` 環境は数式番号が自動で付加される。

表 5-6: ギリシア文字

入力	出力	入力	出力	入力	出力	入力	出力
\alpha	α	\eta	η	\nu	ν	\tau	τ
\beta	β	\theta	θ	\xi	ξ	\upsilon	υ
\gamma	γ	\iota	ι	\o	\o	\phi	ϕ
\delta	δ	\kappa	κ	\pi	π	\chi	χ
\epsilon	ϵ	\lambda	λ	\rho	ρ	\psi	ψ
\zeta	ζ	\mu	μ	\sigma	σ	\omega	ω

注) コマンドの先頭を大文字ではじめるとギリシア文字の大文字になる。

```

1: 二次関数の一般形は
2: \[
3: f_{\{1\}}(x)=x^{\{2\}} + ax + b
4: \]
5: となる。ここで、$-a/2$ を軸、$b$ を
6: $y$-切片と呼ぶ。
7: $f_{\{1\}}(x)$ を基本変形すると
8: \begin{eqnarray}
9: f_{\{1\}}(x)
10: &=& (x^{\{2\}} + ax) + b \\
11: &=& (x + \frac{a}{2})^2 - \frac{a^2}{4} + b
12: -\frac{a^2}{4} + b
13: \end{eqnarray}

```

— (数式環境の出力例) —

二次関数の一般形は

$$f_1(x) = x^2 + ax + b$$

となる。ここで、 $-a/2$ を軸、 b を y -切片と呼ぶ。

$f_1(x)$ を基本変形すると

$$f_1(x) = (x^2 + ax) + b \quad (1)$$

$$= (x + \frac{a}{2})^2 - \frac{a^2}{4} + b \quad (2)$$

5.2 数式のコマンド

代表的な数式のコマンドを以下に挙げる。すべて数式環境の中のみで有効である。

分数 分数は、\frac{分子}{分母} の形で入力する。

$$\frac{1}{x-2} - \frac{1}{x} = \frac{x-2}{2x} \Rightarrow \frac{1}{2} - \frac{1}{x} = \frac{x-2}{2x}$$

空白 数式環境で空白を入れたい場合は、\, を使う。広い空白をいれたいときは\quad または\quadquad を使う。!\, は、マイナスの空白である。

上付き、下付き 累乗などの上付き文字は、`\{ \}` を使う。添字などの下付き文字は、`_\{ \}` を使う。

$$\text{a}_{n+1} = \text{a}_n^{2+n+1} \quad (\text{quad } n \geq 0) \Rightarrow a_{n+1} = a_n^2 + n + 1 \quad (n \geq 0)$$

和、積 和記号 Σ を使う場合、`\sum_{\text{初期条件}}^{\text{終了条件}}` となる。積記号 \prod は `\prod` で表示される。

$$\begin{aligned} \sum_{i=1}^n i^2 &= \frac{n(n+1)(2n+1)}{6} \\ \zeta(s) &= \prod_{p:\text{素数}} (1 - p^{-s})^{-1} \end{aligned}$$

三角関数など 良く使う関数は、既に専用のコマンドが用意されている。

$$\begin{aligned} \sin x, \cos y, \tan z &\Rightarrow \sin x, \cos y, \tan z \\ \exp(w), \log t, \log_{10} X &\Rightarrow \exp(w), \log t, \log_{10} X \end{aligned}$$

微分、積分 積分記号は `\int`, 微分記号は分数のやりかたを用いて `\frac{d}{dx}` とするかまたは、`\partial` を使う。

$$\begin{aligned} \int \frac{f'(x)}{f(x)} dx &= \log f(x) + C \Rightarrow \int \frac{f'(x)}{f(x)} dx = \log f(x) + C \\ f_{xy} &= \frac{\partial^2}{\partial x \partial y} f \Rightarrow f_{xy} = \frac{\partial^2}{\partial x \partial y} f, \end{aligned}$$

ギリシア文字, その他記号など ギリシア文字の一覧を表 5-6 に示す。その他よく使う記号類を表 5-7 に示す。TeX にはいろいろな記号がたくさん用意されているので、興味がある方は参考文献等を参照されたい。

5.3 数式の参照

数式の参照を行いたい場合、まず数式に式番号をつける必要があるため、`equation` 環境もしくは `eqnarray` 環境を用いる。参照したい数式番号の所に `\label{tags}` を貼り、参照する文中では `\ref{tags}` を使う。`tags` は共通な大小英文字ならなんでもよいが、「環境：内容を連想させる単語」の形がおすすめである。下の例では、`eq:xyint`, `eq:rint` を用いている。なお、参照を利用する際、一度 `aux` ファイルに参照情報を書きだす必要があるため、コンパイルを二回以上行う必要がある。

表 5-7: その他の記号

入力	出力	入力	出力	入力	出力	入力	出力
$\{x\}$	$\{x\}$	\leq	\leq	\equiv	\equiv	\leftarrow	\leftarrow
$\langle x \rangle$	$\langle x \rangle$	\geq	\geq	\neq	\neq	\rightarrow	\rightarrow
\overline{x}	\overline{x}	\in	\in	∞	∞	\Leftarrow	\Leftarrow
$x \pm y$	$x \pm y$	\cdot	\cdot	\times	\times	\Rightarrow	\Rightarrow
$x \div y$	$x \div y$	\dots	\dots	\subset	\subset	\mapsto	\mapsto
\leftrightarrow	\leftrightarrow	\emptyset	\emptyset	\supset	\supset	\circ	\circ

```

1: \begin{equation}
2: B:=\int\int_{[0,\infty)^2} e^{-(x^2+y^2)} dx dy \label{eq:xyint}
3: 
4: \end{equation}
5: 式 (\ref{eq:xyint}) に極座標変換
6: $x=r\cos\theta$, $y=r\sin\theta$ を代入すると、
7: 
8: \begin{eqnarray}
9: B &= & \int_{r=0}^{\infty} \int_{\theta=0}^{\pi} e^{-r^2} r dr d\theta \\
10: && \frac{\pi}{2} \int_0^\infty e^{-r^2} r dr
11: &= & \frac{\pi}{2} \int_0^\infty e^{-r^2} r dr
12: && (4)
13: & & となる。式 (4) を...
14: \end{eqnarray}
15: となる。式 (\ref{eq:rint}) を...

```

— (出力例) —

$$B := \int \int_{[0,\infty)^2} e^{-(x^2+y^2)} dx dy \quad (3)$$

式 (3) に極座標変換 $x = r \cos \theta$, $y = r \sin \theta$ を代入すると、

$$\begin{aligned} B &= \int_{r=0}^{\infty} \int_{\theta=0}^{\pi} e^{-r^2} r dr d\theta \\ &= \frac{\pi}{2} \int_0^\infty e^{-r^2} r dr \end{aligned} \quad (4)$$

となる。式 (4) を...

5.4 場合分け、行列

ここでは `array` 環境の使い方を見てみよう。だいたい、`tabular` 環境と同じなのだが、`array` 環境のほうは数式環境内でしか使えない点が異なる。書式としては

```

\begin{array}{書式フォーマット}
第一項 & 第二項 & .... & 最終項 \\
第一項 & 第二項 & .... & 最終項 \\
...
第一項 & 第二項 & .... & 最終項
\end{array}

```

となる。行列を書くには、この両側に伸縮可能なカッコをつければよい。伸縮可能なカッコは、`\left(...\right)` のように、`\left` と`\right`で指定する。これらは(表示行として)同じ行に対で存在しなければならない。`\right.` で右側の出力を省略できる。

```

1: $n$ 個の$n$変数関数
2: \[
3: \left.\begin{array}{c}
4: y_1 = f_1(x_1, \dots, x_n) \\
5: \dots \\
6: y_n = f_n(x_1, \dots, x_n)
7: \end{array}\right.
8: \right.
9: \]
10: に対し次の行列をヤコビ行列という。
11: \[
12: \left(\begin{array}{cccc}
13: & & & \\
14: \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\
15: \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\
16: \dots & & & \\
17: \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \\
18: \vdots & \vdots & \ddots & \vdots \\
19: \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \\
20: \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \\
21: \dots & & & \\
22: \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \\
23: \end{array}\right)
24: \right)
25: \]

```

— (出力例) —
 n 個の n 変数関数

$$\left\{ \begin{array}{l} y_1 = f_1(x_1, \dots, x_n) \\ \dots \\ y_n = f_n(x_1, \dots, x_n) \end{array} \right.$$

に対し次の行列をヤコビ行列という。

$$\left(\begin{array}{cccc} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{array} \right)$$

6 外部環境の利用

6.1 usepackage

\LaTeX にはすでに説明した基本的な環境の他にも非常に多くの環境が外部パッケージの形で用意されている。それらは、`\usepackage{パッケージ名}` をプリアンブルに書くことで利用できる。

表 6-8: \includegraphics のオプション

指定子	意味
<code>width=length</code>	図の幅を <i>length</i> になるように拡大/縮小する。 <i>length</i> は <code>8cm</code> , <code>22mm</code> , <code>0.8\textwidth</code> (全体の 0.8 倍) のように指定する。
<code>height=length</code>	図の高さを <i>length</i> になるように拡大/縮小する。
<code>scale=num</code>	図を <i>num</i> 倍に拡大/縮小する。
<code>angle=num</code>	図を <i>num</i> 度回転する。
<code>bb=a b c d, clip</code>	図の <i>x</i> 座標 $a \leq x \leq c$, <i>y</i> 座標 $b \leq y \leq d$ の範囲だけを表示する。
<code>clip</code>	描画領域の外側を描かない。

6.2 図

図を文中にとりこみたい場合は `graphicx` 外部パッケージを使う。TeX 自身にも「図を描く」機能も存在するが、通常は他のソフトウェア (UNIX 系の OS ならば `tgif` や `gnuplot` など) を用いて作成した図やグラフを文書中に取りこむ方法がとられる。その際、

`\includegraphics[オプション]{図のファイル名}`

とする。`\includegraphics` のオプションの一覧を表 6-8 に示す。オプションを複数指定するときは、`[scale=0.3, angle=90, clip]` のように、コンマで区切って並べる。

6.2.1 tgif を用いて図を描く

以下 `tgif` の説明をするが、`Mathematica` や `gnuplot` でグラフを `eps` ファイルとして作っていれば、それを利用して小節 6.2.2 に進んでもよい。

`tgif` を起動し、次の手順で簡単な図を作成する。

1. 起動。 `eib-pc001:~> tgif &`
2. 操作。 左側に縦に並んでいる図形を選択して、マウスで操作し図を描く。線の太さや種類を変えたり中を塗りつぶしたりするのは左上から選択。
3. 保存。 左上ツールバーの `File->Save` を選択して、保存するファイル名 (例えば `hoge`) を入力。すると、カレントディレクトリに拡張子 `.obj` のファイル (`hoge.obj`) ができる。
4. TeX に取りこむファイルを作る。 左上ツールバーの `File->Print/Export Format` で `EPS` を選択後 (通常デフォルトで `EPS` に設定されているので、実際は何もしなくてもいい)、`File->Print` を選択。すると `hoge.eps` がカレントファイルに出力される。(Print といってもプリンタに出力されるわけではないので注意。)

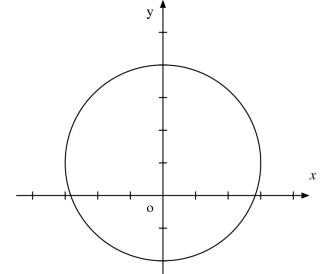
6.2.2 図を取り込む

取り込みたい図(.eps形式のファイル)がTeXソースファイルと同じディレクトリにあることを確認しておく。

```

1: \documentclass[a4j]{jarticle}
2: \usepackage[dvipdfmx]{graphicx}
3: \begin{document}
4: 中心$(0,1)$の円のグラフを描く。
5: \includegraphics[width=6cm]{graph.eps}
```

—(出力例)—



中心 \$(0, 1)\$ の円のグラフを描く。

\includegraphics{...} を直に文中に記述してもよいのだが、通常は `figure` 環境を使い、図番号、図のキャプションとともに表示する。`figure` 環境は `table` 環境と同じくフロートである。

```

1: \documentclass[a4j]{jarticle}
2: \usepackage[dvipdfmx]{graphicx}
3: \begin{document}
4: 図\ref{fig:graph}は $(0,1)$
5: を中心とする円のグラフである。
6: \begin{figure}
7: \includegraphics[width=6cm]{graph.eps}
8: \caption{円のグラフ}
9: \label{fig:graph}
10: \end{figure}
```

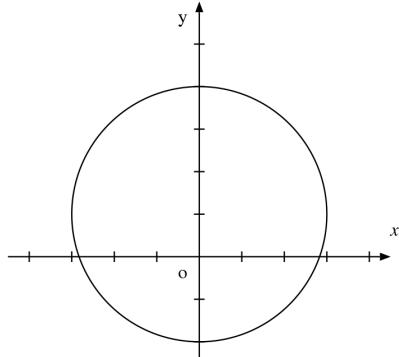


図 1: 円のグラフ

図 1 は \$(0, 1)\$ を中心とする円のグラフである。

数式の参照と同様、図番号の参照も、\caption{} の直後に \label{tags} を貼ることにより、文中で \ref{tags} を使えるようになる。

7 参考文献

TeXでは、参考文献を記述したい場合、`thebibliography` 環境を用いる。この環境内で、文献の項目は次の例のような\item{tags}で指定する。また、文中で文献を引用する場合、`cite` 環境を用いる。参照したい文献を文中の適切なところに\cite{tags}を使う。なお、文献の参照を利用するには、数式の参照と同様に一度aux ファイルに参照情報書き出す必要があるため、コンパイルを二回以上行う必要がある。

```

1: \section*{吾輩は猫である}
2: 「吾輩ハ猫デアル」は夏目漱石の小
3: 説である。俳句雑誌「ホトトギス」
4: の文献\cite{猫}に連載された。
5: 書き出しが次の通り。
6:
7: \begin{thebibliography}{9}
8: \bibitem{猫} 夏目漱石,『吾輩ハ猫デア
9: ル』,「ホトトギス」明治38年2月号,
10: 八巻五号(1905)
11: \end{thebibliography}

```

吾輩は猫である

「吾輩ハ猫デアル」は夏目漱石の小説である。俳句雑誌「ホトトギス」の文献 [1] に連載された。書き出しが次の通り。

参考文献

[1] 夏目漱石,『吾輩ハ猫デアル』,「ホトトギス」明治38年2月号,八巻五号(1905)

実際にレポート等の作成に \TeX を使おうとする場合、適当な本を一冊手元に用意したほうがよい。まず一冊手に入れるなら、バランスのとれている次の本がよいだろう。

奥村晴彦,「 $\text{\LaTeX} 2_{\varepsilon}$ 美文書作成入門」[1].

少し古くなった感があるが、次の本も網羅的でよい。

中野賢,「日本語 $\text{\LaTeX} 2_{\varepsilon}$ ブック」[2].

外部パッケージについてさらに情報が欲しい方には、次の二冊がある。辞書のように使って便利であるが中級者向け。

M. Goossens 他,「The \LaTeX コンパニオン」[3].

M. Goossens 他,「 \LaTeX グラフィックコンパニオン」[4].

Windows 系の OS で \TeX を動かしたい人には、次の書籍、およびその続編がある。

乙部巖己 他,「p $\text{\LaTeX} 2_{\varepsilon}$ for WINDOWS Another Manual Vol.1 Basic Kit」[5].

8 実習操作 その 2

実習 8.1 関数 $f(x) = x^3 - \beta x^2 - x^2 + \beta x$ ($\beta > 1$) について、微分 df/dx を計算した後、ディスプレイに増減表を表示せよ（斜め矢印は \nearrow (\nearrow), \nwarrow (\nwarrow), \searrow (\searrow), \swarrow (\swarrow) を利用しなさい）。

実習 8.2 第 5.3 節の枠囲みの続きを完成させ、 $A = \int_0^\infty e^{-t^2} dt$ の計算結果をディスプレイに表示せよ（ヒント: $A = \sqrt{\pi}$ である）。

実習 8.3 極座標変換

$$\begin{cases} x = r \cos \theta \cos \zeta, \\ y = r \cos \theta \sin \zeta, \\ z = r \sin \theta \end{cases} \quad (r > 0, 0 \leq \theta, \zeta < 2\pi)$$

のヤコビ行列(§5.4 参照)を計算し、その行列をディスプレイに表示せよ。

実習 8.4 以下で与えられる領域 D を作図し、以下の空欄を埋めた後、全体を可能な限り忠実に表示する TeX ソースを作成し、コンパイルとディスプレイへの表示を実行せよ。なお、式番号、図番号には参照(§5.3 節)を利用せよ。

$z = x + yi$ を複素数とする。領域 D は次の不等式：

$$D = \{z = x + yi \mid |z| \leq 1, |z - 1| \geq 1, |z + 1| \geq 1, y > 0\}$$

で定義されているとする。このとき、 D の頂点を求めたい。

$|z| = \sqrt{x^2 + y^2}$ なので、 D の境界は

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

$$\boxed{} = 1 \quad (2)$$

$$\boxed{} = 1 \quad (3)$$

で与えられる。(1)から(2)を引くことにより、

$$2x - 1 = 0$$

を得る。すなわち $x = \frac{1}{2}$ となる。これを(1)に代入すると

$$\left(\frac{1}{2}\right)^2 + y^2 = \frac{1}{4} + y^2 = 1$$

すなわち、 $y^2 = \frac{3}{4}$ となる。 $y > 0$ なので、 $y = \boxed{}$ を得る。同様に、(1)から(3)を引くことにより、

$$\boxed{}$$

を得る。すなわち、 $x = \boxed{}$ となる。これを(1)に代入して、 $(x, y) = (\boxed{}, \boxed{})$ を得る。同様に、(3)から(2)を引くことにより、

$$4x = 0$$

を得る。これを(2)に代入することにより、 $(x, y) = (\boxed{}, \boxed{})$ となる。

以上より、図 1 で与えられる D の各頂点は $z = \boxed{}, \boxed{}, \boxed{}$ の三点で与えられる。

【図を挿入】

9 レポート課題

実習 2.1 以外の各実習操作において(1)作成した TeX ソースコードと(2)出力結果を明示したレポートを作成し、提出しなさい。提出方法、締切日などは担当者の指示に従いなさい。

参考文献

- [1] 奥村晴彦, 「 $\text{\LaTeX} 2\epsilon$ 美文書作成入門」改訂第8版, 技術評論社, 2020.
- [2] 中野賢, 「日本語 $\text{\LaTeX} 2\epsilon$ ブック」, ASCII 出版, 1996.
- [3] M. Goossens 他, 「The \LaTeX コンパニオン」, アスキー出版, 1998.
- [4] M. Goossens 他, 「 \LaTeX グラフィックコンパニオン」, アスキー出版, 2000.
- [5] 乙部巖己 他, 「p $\text{\LaTeX} 2\epsilon$ for WINDOWS Another Manual Vol.1 Basic Kit」, ソフトバンク 1998.