岡山理科大学総合情報学部情報科学科
計算機科学実験: TEX を使った作図

実験担当者編集（１９９８年）

1　課題

実験室では次の各項を学ぶ

1. 実験室のパソコン操作 (電源，キーボード，エディタ，プリンタ)
2. LATEX の使い方
3. 座標変換
4. LATEX による作図

2　LATEX の概要

文書処理システム TEX は Stanford University の Donald Knuth が自らの著作物を印刷するために開発した文書清書システムです。完璧主義の彼は，自著 The Art of Computer Programming シリーズ中の印刷ミスを全て無くしたいと考え，学生に呼びかけ，ミス 1 箇所発見につき懸賞金 5 ドルを出してさえいました。そして最新鋭の計算機環境の中に居た彼はついに，印刷所でのタイプミスを減らすことや仕上がりを自分で確かめたいなどの理由から，コンピュータを操作しさえすれば，ページのレイアウトはもちろん数式や文字を，美しく印刷するシステム TEX を作ってしまったのです。

TEX は，数式の表現能力が高いので，コンピュータ科学者ばかりでなく数学者や自然科学を研究する人々に広く使われはじめました。しかし，かなり細かい指定をしないと思った結果が得られないという人もいました。そんな状況の中で，Leslie Lamport はより簡単な指定だけで，ある程度の文書が作成できる TEX のマクロを作り上げたのです。そして自分の名前の最初の 2 文字を付けてこのシステム名を LATEX としました。この実験の目的は，この LATEX を使って図入りの技術文書を作るというものです。また，簡単な多面体をより立体的に見えるように描くためには回転や平行移動等の座標変換をどのように行えば良いのかについても実験します。

さて，美しい文字を使いレイアウトを工夫した印刷物を作成するにはワードプロセッサが使われることが多いようです。パソコンを利用したソフトでは「ワード」が最も有名だと思われますが，このようなワードプロセッサと TEX の違いを表**??**のようにまとめてみました。

なお，TEX の他に，コマンド型で文書を整形するものには **roff** や今流行の WWW のデータを記述する (**HTML, SGML**) などがあります。表にも示してありますが，これらの良いところは，プリンタや画面表示のためのビューワをそれぞれ自由に組み合わせて選べることです。そして，簡単なプリンタでも印刷所で使う写植機でも，同じデータをそのまま利用することができます。

	ワードプロセッサ	TEX
処理形態	WYSIWYG:(What you see is what you get) つまり見た通りの編集と印刷ができる。	BATCH:テキストにコマンドを埋め込み，一括処理する。レイアウトはプレビューワや印刷により確認。
動作環境	それぞれのハードウェアに専用のソフトウェアが必要。	パソコン，ワークステーション等ほとんどのコンピュータで動作。様々な解像度のプリンタに対応。
使い易さ	画面で見た通りにプリントされるので，理解が容易。短い文書に向いている。	レイアウトを気にせずに文章をどんどん入力できる。専用のコマンドを覚えなければならないが，レイアウトの変更が自由。
長所	専用のキー割当や機能があり簡単で使い易い。入力から印刷までの処理の流れが理解し易く直感的。	複雑な数式を美しく印刷できる。書き方さえ覚えれば，いろいろなマシンや OS で利用できる。抽象的なので，ファイルサイズが小さい。
難点	複雑な数式が表現しにくい。ファイルサイズが大きく，大量の文書処理は苦手。応用が利かない。	入力時に出来上がりがわからない。ある程度コマンドを覚えるまで使えない。

表 1: ワードプロセッサと TEX の比較

3 LATEX の処理の流れ

LATEX は，先にも書いたように BATCH 型の文書処理システムなので，図**??**のような手順

で作業を行う。まず通常のテキストエディタでソースファイル (TEX のコマンドと文章) を作成する。このファイル名を例えば xxxxxx.TEX とする。次にこのファイルを LATEX で処理する。エラーがなければ xxxxxx.DVI というファイルが作成される。エラーの場合はソースを修正する。でき上りをプレビューワで確認し，良ければ印刷し，そうでなければ再度エディタを使い xxxxxx.TEX を修正する。実習に用いる TEX では統合環境の中でこれらの処理を行う。

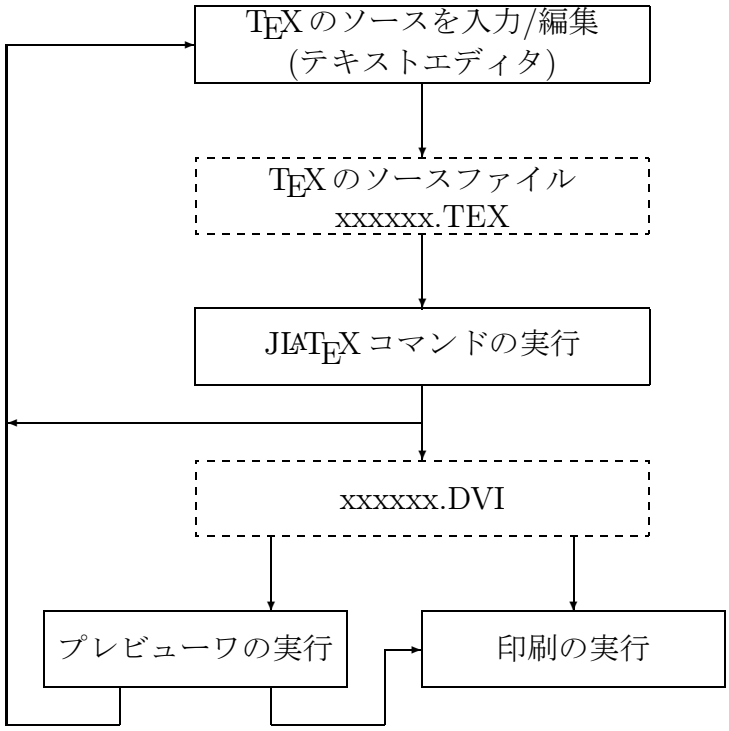


図 1: LATEXの処理の流れ

4　パーソナルコンピュータの操作

実験室のパーソナルコンピュータを使い，LATEXによる処理を次のように行う。

パソコンの電源 ON ⇒ Ctrl+Alt+Del を押してください

⇓

Ctrl+Alt+Del を同時に押す

⇓

ユーザ名，パスワード入力画面が表示される

⇓

ユーザ名，パスワードを入力し，OK をクリックする

⇓

しばらくして Windows の画面が表示される

⇓

「WinTeX32」と書かれたアイコン

をダブルクリックする

⇓

この画面を通して，図??に示したTEXの一連の作業を行う統合環境WinTeXが使える。ウィンドウの上から2行目に並んでいる七つの文字列はメニューバーというもので，それぞれの文字列をマウスで1回クリックするといくつかの機能がメニュー表示される。以下にそれぞれのメニューを説明する。

ファイルは，TEXやLATEXのソースファイルを作成や編集するときに使うテキストエディタを起動する場合に使う。メニューから「新規作成」か，以前作成したものを修正・追加する場合は「開く」をクリックする。

- 「新規作成」を選んで作成したソースに対しては，「TeXをかけてプレビュー」を実行すると，保存するためのファイル名を聞いてくる。ドライブにz:を指定し，適切なファイル名 (xxx.tex) を指定しサーバのディスクに保存する。

フロッピーディスクに保存したい場合は，ドライブa:に適切な名前をつけたファイル名で保存する。保存する際のファイル名はどう決めようと自由ではあるが，自分のアルファベット名の最初の4文字程度に数字を01,02などと付けて区別するようにしておくと以降の作業効率が良くなる：例えば`TAROO1.TEX`などとする。

プロジェクトは，上で作ったソースファイルを処理するときに使う。通常は上から三つまでの機能「TeXをかける」「TeXをかけて印刷」「TeXをかけてプレビュー」を使う。特に，「TeXをかけてプレビュー」はTeX処理 (WpvTeX) とプレビュー (Windvi) を連続して実行するので，文章作成時はよく使う。

ソースが構文上正しければプレビュー画面が表示される。エラーの時はWpvTeXのウィンドウにエラーメッセージが表示され，応答待ちになる。通常は**E**を入力してエディタに戻る。そうしたくない時は**X**を入力する。それでもダメならばCtrl+CとかCtrl+Zを押して

みる。それでもまだダメな場合には，タスクマネージャを起動して不要な処理を強制的に中断させることもできる (要注意)。なお，用紙の無駄遣いを避けるためにも，プレビューワで十分に仕上がりを確認してから印刷するべきである。

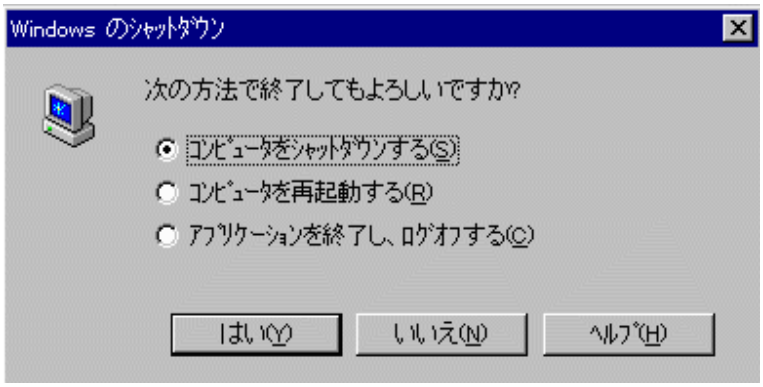
プレビューワで確認後，Windvi ウィンドウのプリントボタン (左から 2 番目) を押し，プリンタに印刷する。印刷ページを指定して必要なページのみを印刷することもできる。プリンタは常時通電してあり，実験室の利用者なら誰でも何時でも利用できるが，誰が印刷したものか分かるような工夫をしておかねばならない。

終了するには，ウィンドウ Windvi, WpvTeX, WinTeX32 を順次閉じる。ウィンドウを閉じるには右上角の

をクリックする。

システムを終了するには，画面左下の

をクリックし，次にシャットダウンをクリックすると「コンピュータのシャットダウン」というメッセージボックスが出てくる。



この中から はい (Y) をクリックする。

コンピュータは自動的にシステムを終了させて電源を OFF してくれる。したがって，

コンピュータが動作中は電源ボタンを押さないこと。

5 LaTeX の書き方

全体構成

LaTeX の文書は次のような構成をしていなければならない。

```
\documentstyle[オプション]{スタイルファイル名}
プリアンブル（ページ書式の細かい設定など）
\begin{document}
本文
\end{document}
```

ここで，**オプション** と**スタイルファイル名** で指定する主なものは次の通り。

オプション	意味
11pt	11 ポイントで印刷
12pt	12 ポイントで印刷
twocolumn	文書を 2 段組で印刷

スタイルファイル名	用途
jarticle	日本語論文
jreport	日本語のレポート
jbook	日本語の本

プリアンブルはページレイアウトを特に変更する時に使うので，各スタイルファイルの標準値を使う場合は何も書かなくてよい。
本文は普通の文書ならば，ずらずらと続けて書いてよい。どこで改行しようと，LaTeX で処理すれば綺麗にそろえてページ内に納めてくれる。段落を変えるには，\par というコマンドを入れてやる。以上の説明だけで，通常の文書は LaTeX で作成することができる。

文書の標題出力

論文等に付ける標題は，\documentstyle の直後に次のように指定するとよい。

```
\title{標題名}
\author{著者名}
\date{日付}
\maketitle
```

これにより，このテキストの表紙のように印刷される。

特殊記号

次の記号は LaTeX ではコマンド等の特殊な意味を持つので，記号そのままの文字を入力する場合と区別しなければならない。

その他の記号として，クォーテーションの開きと閉じを区別して使うように気をつけねばならない。‘‘Hello!’’ と入力すると “Hello!” と表示され,。‘Hello!’ と入力すると ‘Hello!’ と表示される。

フォントの種類と大きさ

フォントの大きさと，フォントの種類を指定するコマンドをその例を示して表にまとめた。これらの指定は，コマンドが書かれた以降の文字すべてに適用される。戻したい場合は，そこに\normalsize や\rm ，\mc などのコマンドを入れる。また，一部の文字だけに適用したい場合は，{\it italic } のようにすると *italic* だけがイタリック体になる。

文字の大きさ指定

表 2: 特殊な意味を持つ文字

特殊記号	LaTeX での意味	この文字の表示
\	LaTeX のコマンドの開始	$\backslash$
{, }	制御範囲の指定	\{ , \}
$	数式モードの開始，終了	\$
&	タブ文字	\&
#	マクロ引き数	\#
^	上付の添え字	\^{}
_	下付の添え字	_{}
%	コメント行	\%
~	改行しない 1 文字空白	\~{}

コマンド	大きさ
\tiny	tiny 相当小さい
\scriptsize	scriptsize かなり小さい
\footnotesize	footnotesize ちょっと小さい
\normalsize	normalsize 普通のサイズ
\large	large ちょっと大きい
\Large	Large かなり大きい
\LARGE	LARGE 相当大きい
\huge	huge でかい
\Huge	Huge ドデカイ

文字の種類の指定

コマンド	表示される文字の種類
\rm	Roman face ローマン
\bf	**Bold face** ボールド
\it	*italic face* イタリック
\sl	*Slanted face* 斜体
\sf	Sans serif face サンセリフ
\sc	Small caps face 小大文字
\tt	`Typewriter face` タイプライタ
\gt	**日本語ゴシック**
\mc	日本語明朝

位置寄せ

位置寄せには次の3種類がある。それぞれ \\を行の最後に付けることで複数行について指定することができる。

\begin{center}
この文字が中央に配置される\\
その下に，これも中央に
\end{center}

\begin{flushright}
この文字が右寄せされる
\end{flushright}

\begin{flushleft}
この文字が左寄せされる
\end{flushleft}

箇条書き

先頭に黒丸が付くものと，数字が付く箇条書きの指定方法は次の通りである。これらは，入れ子にもすることができる。

黒丸の箇条書き

```
\begin{itemize}
  \item 項目A
  \item 項目B
\end{itemize}
```

数字の付いた箇条書き

```
\begin{enumerate}
  \item 項目1
  \item 項目2
\end{enumerate}
```

スペースと区切り

\vspace{長さ}	縦方向の空白。長さは 5mm などと書く
\hspace{長さ}	横方向の空白。長さは 5mm などと書く
\\	改行
\par	段落の終了(次の行が字下げ[インデント]される)
(空行)	同上
\noindent	上記コマンドで改行した後に指定すると，字下げされない
\newpage	改頁

段落

LaTeX では，自動的に段落番号を付けてくれる機能がある。

```
\chapter{章}  (book スタイルしか使えない)
\section{節}
\subsection{小節}
\subsubsection{小小節}
```

6　プリアンブル

プリアンブルは，各ページのヘッダ，フッタや文字印刷領域の設定を標準値と変えるときに指定する。次節の例で\documentstyle と\begin{document} の間にある5行がプリアンブルの指定例である。ここでは，マージン(余白)と本文領域を指定している。

付録に設定パラメータとその標準値を示すので，適時参照して欲しい。

7　表紙

スタイル jarticle に標準の表題作成コマンドは節「LaTeX の書き方」にある指定を\documentstyle の直後に書くことで簡単に作成できる。ただし，この場合の表題は独立した 1 ページにはならないで，本文がすぐ下に続く形になる。このテキストの例を次にあげる。

```
\documentstyle[12pt]{jarticle}

\topmargin 0cm
\evensidemargin 0cm
\oddsidemargin 0cm
\textheight 23cm
\textwidth 16.5cm

\begin{document}
\title{計算機科学実験: \TeX を使った作図}
\author{実験担当者}
\date{}
\maketitle

\section{課題}
実験室では次の各項を学ぶ
\begin{enumerate}
  \item 実験室のパソコン操作(電源，キーボード，エディタ，プリンタ)
  \item \LaTeX の使い方
  \item 座標変換
  \item \LaTeX による作図
\end{enumerate}
```

⋮

この指定と，今読んでいる表紙そのものを見比べてみよう。3行目の\topmargin から7行目の\textwidth まではプリアンブルの記述である。次の\begin{document}から\maketitle の記述を見てみる。ここの指定により，実習テキストの表題が表示される。このままでは，レポートの表紙に1ページを割り当てられないし，その他実習日等の指定もできない。そこで，前に作成した表紙を1ページ分使う指定を次のようにして行う。

```
\documentstyle[12pt]{jarticle}
(上の例のプリアンブル指定をそのまま指定する)
\begin{document}
\begin{titlepage}
前に作った表紙の指定
\end{titlepage}
レポートの本文
\end{document}
```

この指定を用いて，各自，自分用の表紙を作成してみよう。

8　表の作成

LaTeX では表の作成のために**tabbing** と **tabular** の二つの環境が用意されている。

tabbing 環境

簡単な位置決めをするだけの表は**tabbing** 環境で作成できる。tabbing 環境の指定方法は次の通りである。tabular 環境と一番違う点は，各列の幅が決まっているので，入れる文字列が長いと次の列にはみ出して書かれる点である。この欠点を逆用するとプログラムリストのような先頭位置のみをそろえて字下げした文書を効率良く書くことができる。

```
\begin{tabbing}
第1列の幅指定 \= 第2列の幅指定 \= ... 第n列の幅指定 \kill
文字列(1,1) \> 文字列(1,2) \> ... 文字列(1,n) \\
...
文字列(m,1) \> 文字列(m,2) \> ... 文字列(m,n)  ※最後の行は\\を付けない
\end{tabbing}
```

ここで，各列の幅指定には次の2通りの方法がある。

1. **ダミー文字指定**: 同列で最大の長さの文字列より長い文字列を指定する。例えば，xxxxxxxxxx のように x を適当な長さになるように書く。

2. **\hspace 指定**: \hspace{3cm} のように，長さを明示的に指定する。

例

```
\begin{tabbing}
XXXXXXXXXXXXX \= XXXXXX \= XXXXXXXXXXXXXXXX \kill
{\it Architecture} \> {\it Chip} \> {\it Maker} \\
ARM \> ARM 6/7 \> VSLI Technology \\
PowerPC \> PowerPC 6xx \> IBM, Motorola \\
x86 \> Pentium \> Intel \\
MIPS \> R4000/4400 \> IDT, NEC \\
HP-RISC \> PA-RISC7xxx \> HP
\end{tabbing}
```

これを，LaTeX で処理すると次のように印刷される。

Architecture	*Chip*	*Maker*
ARM	ARM 6/7	VSLI Technology
PowerPC	PowerPC 6xx	IBM, Motorola
x86	Pentium	Intel
MIPS	R4000/4400	IDT, NEC
HP-RISC	PA-RISC7xxx	HP

この例では，2番目の幅が小さ過ぎたため3番目の列に重なってしまっているところがある。XXXXXXXX となっているところを `\hspace` で置き換えて表を作成してみよう。

次にプログラムリストをこの環境で書いてみる。列幅にはインデントの標準値として割り付けられている横方向の空白`\quad`を使うことにする。

```
\begin{tabbing}
\quad \= \quad \= \quad \= \quad \= \quad \kill
{\bf void} $bubblesort({\bf int}~n,~{\bf keytype}~a[~])$ \\
\{ \\
 \>  {\bf int} $i, j, k;$ \\
 \>  {\bf keytype} $x;$ \\
\\
 \>  $k = n - 1;$ \\
 \>  {\bf while}$(k >= 0) \{$ \\
 \> \>    $j = -1;$ \\
 \> \>    {\bf for}$(i = 1; i <= k; i++)$ \\
 \> \> \>      {\bf if}$(a[i - 1] > a[i]) \{$ \\
 \> \> \>  \>       $j = i - 1;$ \\
 \> \> \>  \>       $x = a[j];~ a[j] = a[i];~ a[i] = x;$ \\
```

```
 \> \> \>       \} \\
 \> \>      $k = j;$ \\
 \>  \} \\
\}
\end{tabbing}
```

これを LaTeX で処理すると次の結果が得られる。この例で $ と $ に囲まれているところは，数式モードといい，文字をイタリックにしたり数式を綺麗に表示できる。ただし，英文字と英文字の間の空白を詰めてしまうので，明示的に空白コマンド~を入れなければならない。

void *bubblesort*(**int** n, **keytype** $a[\,]$)
{
  int i, j, k;
  keytype x;

  $k = n - 1$;
  while$(k >= 0)$\{
    $j = -1$;
    for$(i = 1; i <= k; i++)$
      if$(a[i-1] > a[i])$\{
        $j = i - 1$;
        $x = a[j];\ a[j] = a[i];\ a[i] = x$;
      \}
    $k = j$;
  \}
}

そのまま表示する環境 verbatim

ここでは，表作成と直接関係無いがプログラムリストのように入力した文字列が改行も含めて，そのまま表示できる環境を示す。この環境は\begin{verbatim}と\end{verbatim}で挟まれた行に書かれた文字を，空白の数や改行，それに LaTeX で扱われる特殊コマンドも含めてすべてそのまま表示する。環境の性質から，文字タイプは選べない。英字はタイプライタ文字，日本語は明朝体のみである。

```
\begin{verbatim}
void␣bubblesort(int␣n,␣keytype␣a[])
{
␣␣␣int␣i,␣j,␣k;
␣␣␣keytype␣x;

␣␣␣k␣=␣n␣-1;
```

```
␣␣␣while(k␣>=␣0)␣{
␣␣␣␣␣␣j␣=␣-1;
␣␣␣␣␣␣for(i␣=␣1;␣i␣<=␣k;␣i++)
␣␣␣␣␣␣␣␣if(a[i␣-␣1]␣>␣a[i])␣{
␣␣␣␣␣␣␣␣␣␣j␣=␣i␣-␣1;
␣␣␣␣␣␣␣␣␣␣x␣=␣a[j];␣a[j]␣=␣a[i];␣a[i]␣=␣x;
␣␣␣␣␣␣␣␣}
␣␣␣␣␣␣k␣=␣j;
␣␣␣}
}
\end{verbatim}
```

これを LaTeX で処理すると次のように印刷される。

```
void bubblesort(int n, keytype a[])
{
   int i, j, k;
   keytype x;

   k = n -1;
   while(k >= 0) {
      j = -1;
      for(i = 1; i <= k; i++)
         if(a[i - 1] > a[i]) {
            j = i - 1;
            x = a[j]; a[j] = a[i]; a[i] = x;
         }
      k = j;
   }
}
```

tabular 環境

tabular 環境では中に書き入れる文字列の長さに合わせて各列の幅を調整してくれるので，幅の長さを指定する必要はない。また，罫線付の表を作るのも tabular 環境で行うと良い。指定方法は次の通りである。

```
\begin{tabular}{各列の様式指定}
文字列(1,1) & 文字列(1,2) & ... 文字列(1,n) \\
...
文字列(m,1) & 文字列(m,2) & ... 文字列(m,n)
\end{tabular}
```

※最後の行は \\ を付けない。ただし一番下に罫線を引くときは\\を付けた後に \hline を付ける。

各列の様式指定では次のコマンドを使うことができる。これらは指定したコマンド列の左から順にl,c,rのいづれかが現れた順に1列，2列... のものと解釈される。

コマンド	意味
|(ストローク)	表の高さ一杯の縦罫線を引く
||	表の高さ一杯の2重縦罫線を引く
l(エル)	列の文字列を左寄せにする
c	列の文字列を中央にする
r	列の文字列を右寄せにする
p{列の幅指定}	列の幅を指定する
@{文字列}	列のすべてに共通して埋めこむ文字列
*{n}{指定コマンド}	n個の指定コマンドが並んだものと解釈する

例

```
\begin{tabular}{|l|l|l|}
\hline
{\it Architecture} & {\it Chip} & {\it Maker} \\ \hline \hline
ARM & ARM 6/7 & VSLI Technology \\ \hline
PowerPC & PowerPC 6xx & IBM, Motorola \\ \hline
x86 & Pentium & Intel \\ \hline
MIPS & R4000/4400 & IDT, NEC \\ \hline
HP-RISC & PA-RISC7xxx & HP \\ \hline
\end{tabular}
```

これを，LaTeXで処理すると次のように印刷される。

Architecture	*Chip*	*Maker*
ARM	ARM 6/7	VSLI Technology
PowerPC	PowerPC 6xx	IBM, Motorola
x86	Pentium	Intel
MIPS	R4000/4400	IDT, NEC
HP-RISC	PA-RISC7xxx	HP

ここで，\hlineは横罫線を引くためのコマンドで，先頭と各行の後に指定している。1行目の後は2回指定しているので2重線が引かれる。

pコマンドで列の幅を指定すると，入りきらない文字列は折り返して納めてくれる。次の例は，テキストの2ページに印刷された表のソースの一部である。入力と出力の関係を確かめてみよう。なお，同じ枠内で改行したいときは\\は行の終端で使えないので，\parを使うことになる。

```
\begin{table}[htb]
\begin{center}
\begin{tabular}{|l|p{5cm}|p{5cm}|}
\hline
        &\hfil ワードプロセッサ \hfil & \hfil \TeX \\ \hline
処理形態 &WYSIWYG:つまり見た通りの編集と印刷ができる。
  &BATCH:テキストにコマンドを埋め込み，一括処理する。
  レイアウトはプレビューワや印刷により確認。\\
  \hline
動作環境 &それぞれのハードウェアに専用のソフトウェアが必要。
  &パソコン，ワークステーション等ほとんどのコンピュータで動作。
  様々な解像度のプリンタに対応。\\
  \hline
\end{tabular}
\caption{ワードプロセッサと\TeX の比較}
\label{tbl1}
\end{center}
\end{table}
```

9 minipage環境

文書領域の全体ではなくて，ある大きさの枠に文章を埋め込みたいことがある。そのような時にはminipage環境を使うとよい。[ ]内は横に並ぶ文字(行) との位置関係を決めるもので，指定しないと中央が，tを指定すると上端が，bを指定すると下端が横一行とそろった位置に表示される。つまり，minipage で指定した枠も1行の中の大きな文字のように扱われるということである。ただし，次の例のような表等の枠組みがある物は，その中での位置決めパラメータも指定する必要がある。

{ }には枠の横幅を指定する。

例

文字の位置

Architecture	*Chip*	*Maker*
ARM	ARM 6/7	VSLI Technology
PowerPC	PowerPC 6xx	IBM, Motorola
x86	Pentium	Intel
MIPS	R4000/4400	IDT, NEC
HP-RISC	PA-RISC7xxx	HP

このminipageの枠は位置関係のパラメータを指定していないので，先頭の”文字の位置”の行に対して，この小さな文字で書いている枠の，上下方向で見た 中央の位置を揃えて表示する。

入力ソース

```
文字の位置 \quad
\begin{minipage}[t]{9cm}
   \begin{tabular}[t]{|l|l|l|}
   ここには，先ほど入力したtabularの例をコピーアンドペーストで持ってくる。
   \end{tabular}
\end{minipage}
\quad
\begin{minipage}{3cm}
   \scriptsize
   この{\tt minipage}の枠は位置関係のパラメータを指定していないので,
   先頭の"文字の位置"の行に対して，この小さな文字で書いている枠の，上下方向で見た　中央の位置を揃えて表示する。
\end{minipage}
```

10　画像の貼りこみ

画像の貼りこみ方は，オペレーティングシステムによって異なっている。ここでは TeX for Windows を使って画像を貼りこむ方法を説明するが，UNIX や Mac，DOS 等では，ここで説明した通りには使えないので注意しなければならない。ただし，どのシステムであっても，似たような方法で画像を貼り付けることができる。

1. 画像ファイルを用意する。TeX for Windows では Windows 標準画像形式 bmp で作成する。Windows のペイントブラシで作ったものもこの形式である。(UNIX や Mac では PICT や EPS という形式が使われることが多い。)

2. 先頭行に指定する`\documentstyle`の`[`と`]`の中に`hyper`を指定する。例えば，`\documentstyle[a4j,12pt,hyper]{jarticle}`となる。

3. 画像を貼りこみたい位置に`\bmpfile(15cm,8cm){campus1.bmp}` といった記述をする。ここで，`(15cm,8cm)`の二つの数字は表示する画像の横方向と縦方向の長さを示す。元の画像の縦横比と異なるときにはこの比率になるまで伸ばして表示する。`campus1.bmp`は画像ファイル名である。なお，このファイルは本文と同じディレクトリになければならない。

```
\documentstyle[a4j,12pt,hyper]{jarticle}
\begin{document}
        文章
\bmpfile(15cm,8cm){campus1.bmp}
        文章
\end{document}
```

上で示したコマンドにより次に示す図が張り込まれる。

11　ページレイアウトパラメータ

1段組の場合

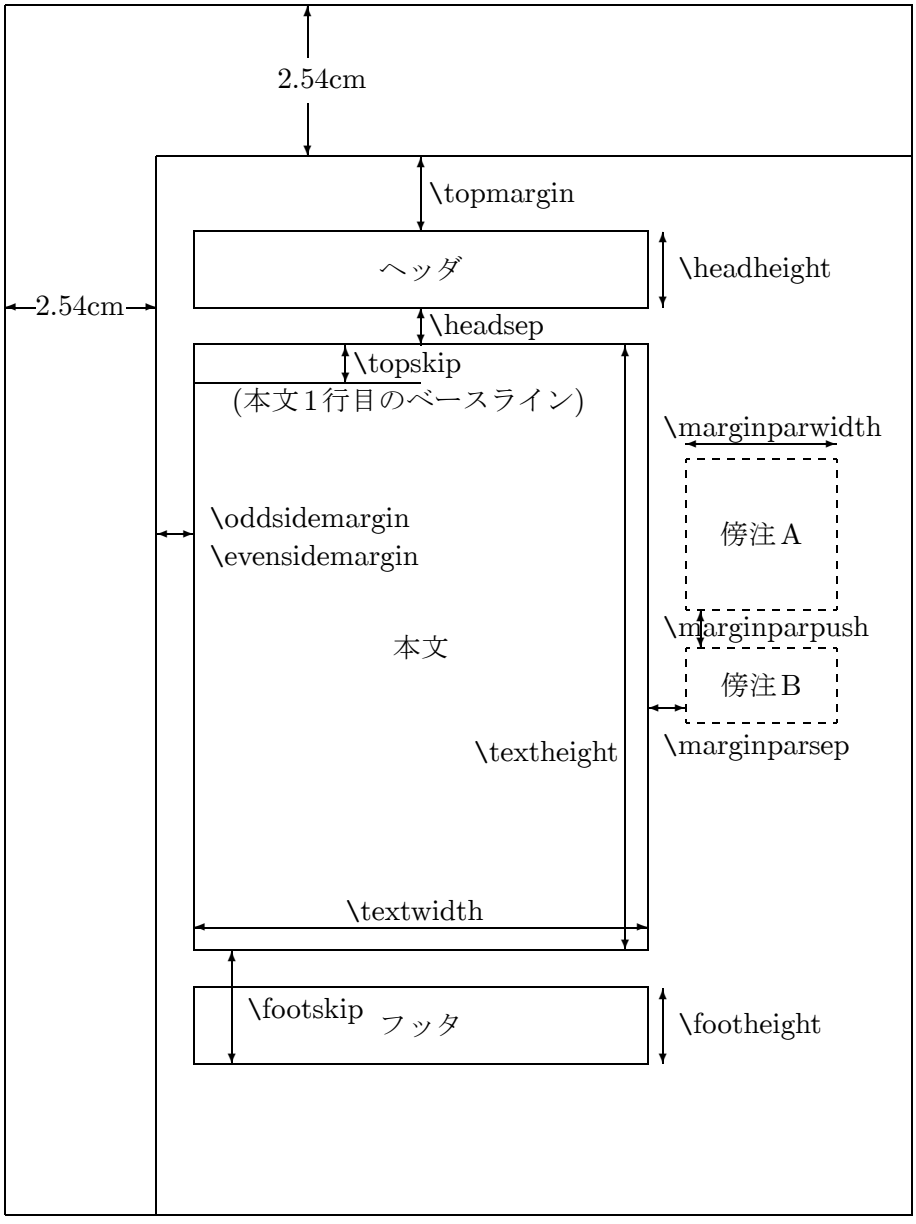


図 2: ページレイアウトパラメータ (1段組)

2段組の場合

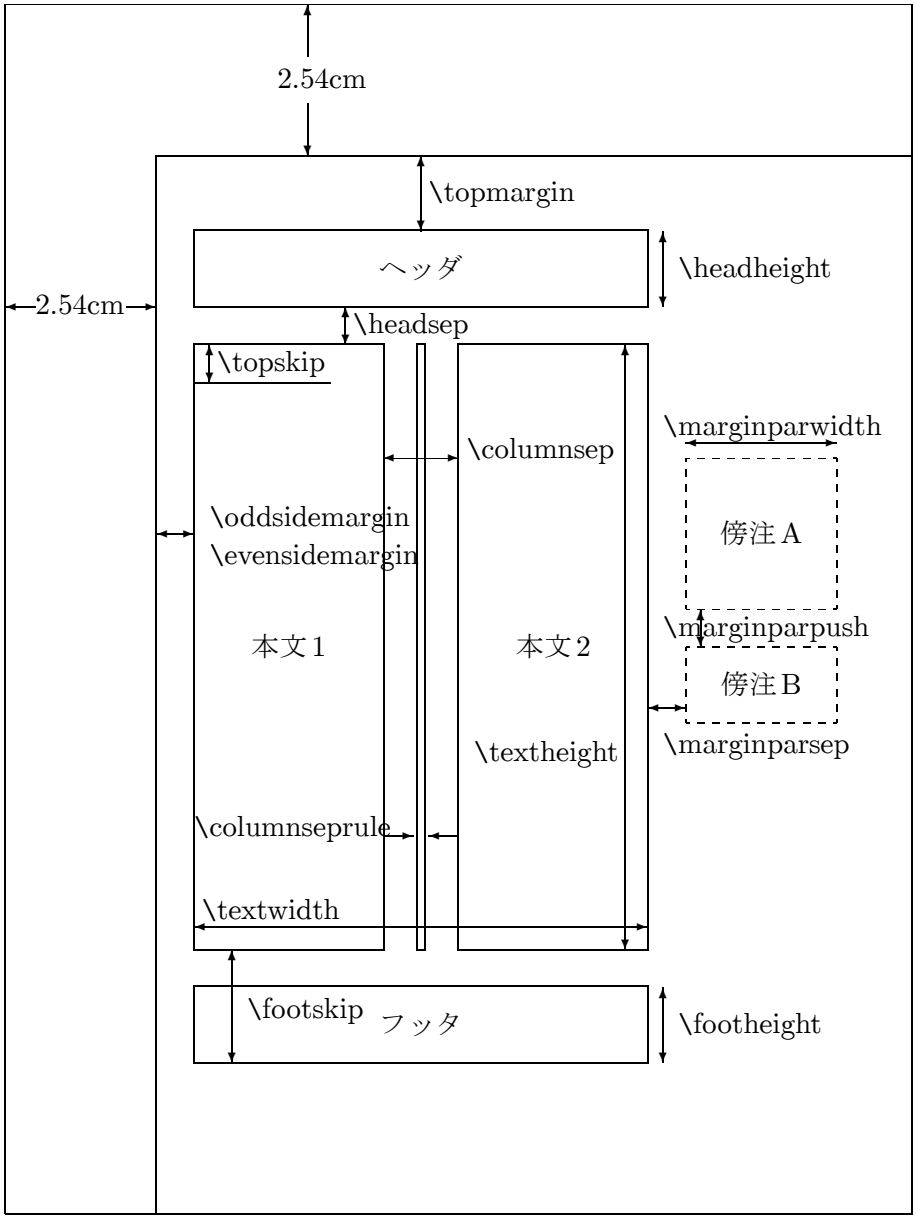


図 3: ページレイアウトパラメータ (2段組)

レイアウトパラメータの解説

図**??**と図**??**で示した各スタイル指定に対応するレイアウトパラメータの標準値を一覧表にして示す。最初の表が1段組を，次の表が2段組を指定した場合の標準値を表している。これらのパラメータを文先頭プリアンブル部で明示的に指定して独自のレイアウトにすることができる。通常は，標準値によるレイアウトで十分用が足りるようになっているが，レイアウトの仕上げに微調整が必要であると感じたら，これらの資料を参考にして挑戦して見られたい。

	article			book			report		
parameter	10pt	11pt	12pt	10pt	11pt	12pt	10pt	11pt	12pt
\oddsidemargin	63pt	54pt	39.5pt				63pt	54pt	39.5pt
(twoside)	44pt	36pt	21pt	0.5in	0.25in	0.25in	44pt	36pt	21pt
\evensidemargin	63pt	54pt	39.5pt				63pt	54pt	39.5pt
(twoside)	82pt	74pt	27pt	0.75in	0.73in	0.73in	27pt	27pt	27pt
\topmargin	27pt	27pt	27pt	0.75in	0.73in	0.73in	27py	27pt	27pt
\headheight	12pt	12pt	12pt	12pt	12pt	12pt	12pt	12pt	12pt
\headsep	25pt	25pt	25pt	0.25in	0.275in	0.275in	25pt	25pt	25pt
\topskip	10pr	10pt	10pt	10pr	10pt	10pt	10pr	10pt	10pt
\textheight	526pt	526.8pt	532pt	502pt	526.8pt	532pt	526pt	526.8pt	532pt
\textwidth	345pt	360pt	390pt	4.5in	5in	5in	345pt	360pt	390pt
\footheight	12pt	12pt	12pt	12pt	12pt	12pt	12pt	12pt	12pt
\footskip	30pt	30pt	30pt	0.35in	0.38in	30pt	30pt	30pt	30pt
\marginparwidth	90pt	83pt	68pt	0.75in	1in	1in	90pt	83pt	68pt
\marginparsep	11pt	10pt	10pt	7pt	7pt	7pt	11pt	10pt	10pt
\marginparpush	5pt	5pt	7pt	5pt	5pt	7pt	5pt	5pt	7pt
\parindent	15pt	17pt	1.5em	15pt	17pt	1.5em	15pt	17pt	1.5em
\mathindent	25pt	2.5em	2.5em	25pt	2.5em	2.5em	25pt	2.5em	2.5em

\oddsidemargin	30pt
\evensidemargin	30pt
\textwidth	410pt
\columnsep	10pt
\columnseprule	0pt
\marginparwidth	48pt
\marginparsep	10pt
\parindent	1em
\mathindent	2em

2段組使用時の標準値　　(1pt ≈ 0.35mm)

表 3: レイアウトパラメータ標準値

12　数式表現のための二つの環境

数式は文章中の数式モードと呼ばれている環境の中だけでしか利用できない。そのための方法が二つある。一つは文章中に数式を埋めこむ表現方法と，もう一方は数式を文章とは独立に表現する方法である。前者を文中数式モード，後者を独立数式モードまたは，ディスプレイモードという。

なお，以下の説明では，複雑な数式を書くために { と } で囲まれたものが一つのかたまりとして扱われることに注意して読んで欲しい。

文中数式モード

文章中に数式を書き込むためには，数式を表す文字列を$ と $ で囲んで指定する。

例

```
In this example, it is  shown that how to describe mathematical
formula  in some document, like this
$\sum_{i=1}^{\infty}\frac{1}{i}$.
Do you know what the meaning of this formula?
```

この結果は，次のように印刷される。

In this example, it is shown that how to describe mathematical formula in some document, like this $\sum_{i=1}^{\infty}\frac{1}{i}$. Do you know what the meaning of this formula?

ここで，最初と最後を$ で挟む代わりに\(と \) で挟んでもまったく同様の結果になるので，各自使い易い方を選んで使えば良い。

独立数式モード

数式を文章から独立して書きたいときは，$$ と $$ で囲んで指定をする。

例

```
In this example, it is  shown that how to describe mathematical formula
in a display mode , like this;
$$
\sum_{i=1}^{\infty}\frac{1}{i}.
$$
Do you know what the meaning of this formula ?
```

この結果は，次のように印刷される。

In this example, it is shown that how to describe mathematical formula in a display mode , like this;

$$\sum_{i=1}^{\infty}\frac{1}{i}.$$

Do you know what the meaning of this formula ?

ここで，最初と最後を$$ で挟む代わりに\begin{displaymath} と\end{displaymath} で挟んでもまったく同様の結果になるので，各自使い易いほうを選んで使えば良い。

13　数式中の文字や関数記号

数式では，普段の文章では使うことの少ないフォントを使うことが良くあるが，そのようなフォントにカリグラフ (Calligraph) 文字がある。$a \in \mathcal{A}$ を表示するには，`$a \in \cal A$` と入力すれば良い。これらは，次のようなものである。

$$\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$$

また，ギリシャ文字 $\alpha, \beta, \gamma, \ldots, A, B, \Gamma \ldots$ は$\alpha, \beta, \gamma, \dots, A, B, \Gamma, \dots $，と入力すれば，関数名 $\sin \cos \log$ は$\sin, \cos, \log$ と入力すれば，演算子 $\times, \div, \subset$, は$ \times, \div, \subset $ と入力すれば表示することができる。付録にこれらの文字や記号の指定コマンドを示しておくので，適時必要なときに参照されたい。

また，$\lim_{x \to \infty} f(x)$ のように，記号に添字を付けることができるものもある。例えば，これは `$\lim_{x\to\infty}f(x)$` と書くことで表示される。

数式モードの中では，英字はすべてイタリック体になるので普通の文字を数式モードのなかに書くときは\mboxを使い，次のように入力する。

```
$$
\Gamma(n)=(n-3)!\quad\mbox{when $n$ is an integer.}
$$
```

ここで，\quad は英字2文字程度の空白を入れるコマンドである。数式モードでは基本的に文字と文字の間をすべて詰めてしまうので，このようなコマンドで間隔を空ける。mboxの中は通常の文章領域と同じく空白は空白として扱われる。上の例は，次のように表示される。

$$\Gamma(n) = (n-3)! \quad \text{when } n \text{ is an integer.}$$

数式モードの中で，ボールド体を使いたい時は\mbox と \boldmath を組み合わせて用いる。$\boldsymbol{x} = (a, b)$ は\(\mbox{\boldmath x} = (a,b) \) と入力することにより太字の x を表示することができる。

数式モードで使えるアクセント付の文字は，次のように指定する。

入力	表示	入力	表示
$ \hat x $	$\hat{x}$	$ \check x $	$\check{x}$
$ \tilde x $	$\tilde{x}$	$ \acute x $	$\acute{x}$
$ \dot x $	$\dot{x}$	$ \ddot x $	$\ddot{x}$
$ \grave x $	$\grave{x}$	$ \breve x $	$\breve{x}$
$ \bar x $	$\bar{x}$	$ \vec x $	$\vec{x}$
$ \widehat x $	$\widehat{x}$	$ \widetilde x $	$\widetilde{x}$
$ \widehat {xy} $	$\widehat{xy}$	$ \widetilde {xy} $	$\widetilde{xy}$
$ \widehat {xyz} $	$\widehat{xyz}$	$ \widetilde {xyz} $	$\widetilde{xyz}$

この他，ある範囲に渡り上線や下線を引くには，`\underline{`下線`}`と`$\overline{`上線`}$` を使い $\underline{\text{下線}}$ と $\overline{\text{上線}}$ のように表示することができる。

14　数式表現

この節では具体的な数式表現を習得する。

基本的数式表現

添え字

べき乗などをあらわすために右肩に数式を乗せるにはハット記号 ^ を用いる。2^n は$ 2^n$と書き表す。また，右下に付ける添字にはアンダーバー _ を使う。a_i を表示するには a_i と入力する。
例

入力	表示	入力	表示
$ x^{2n}$	x^{2n}	$ n_{12}$	n_{12}
$ f'_i $	f'_i	$ 2^{n^k} $	2^{n^k}

分数，根号，二項係数

分数の表現は$2/3, \exp(1/2)$のように文章中ではスラシュ / で書けばよい。ディスプレイモードでは次のように書きたい。

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

これは，次のように入力すればよい。

```
$$ x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
```

ここで，分数の表現は次のように指定すればよい。

```
\frac{分子の数式表現}{分母の数式表現}
```

これに対して TEX では次のように指定する。どちらを使っても良い。

```
{分子の数式表現 \over 分母の数式表現}
```

分母や分子に更に分数が入っている場合は，くり返し\frac を適用して

$$\frac{1+\frac{1}{2}}{2}$$ $$ \frac{1 + \frac {1}{2}}{2}$$

このような，結果を得る。ここで，分母や分子のフォントが小さくなる (上の例では分子の 1/2 が小さくなっている) のが避けたい場合は，\displaystyle を使い，次のように入力する。

$$\frac{1+\frac{1}{2}}{2}$$ $$ \frac{1 + \displaystyle{\frac {1}{2}}}{2}$$

根号は，前の例にも出てきたように，\sqrt{2} のように指定すると，$\sqrt{2}$ が表示される。3 乗根以上の表現は \sqrt[n]{2} のように，`sqrt` の直後に [と] で数字を囲んで指定すると，$\sqrt[n]{2}$ という結果が表示される。

分数と根号の含まれた次の数式を各自で表示してみよう。

$$\sqrt[3]{-\frac{b}{2}+\sqrt{\frac{a^3}{27}+\frac{b^2}{4}}}+\sqrt[3]{-\frac{b}{2}-\sqrt{\frac{a^3}{27}+\frac{b^2}{4}}}$$

これは，次のソースによる。

```
$$
\sqrt[3]{-\frac b2 + \sqrt{\frac{a^3}{27} + \frac{b^2}4}}
+ \sqrt[3]{-\frac b2 -\sqrt{\frac{a^3}{27} + \frac{b^2}4}}
$$
```

２項係数は上下に並んだ式を括弧で括った形をしている。`$ n \choose k$` と書くことで $\binom{n}{k}$ を得る。類似の表現を次に示す。

$$\binom{n}{k} \qquad \texttt{\$\$ \{n \choose k\} \$\$}$$

$$\left\{ n \atop k \right\} \qquad \texttt{\$\$ \{n \brace k\} \$\$}$$

$$\left[n \atop k \right] \qquad \texttt{\$\$ \{n \brack k\} \$\$}$$

$$n \atop k \qquad \texttt{\$\$ \{n \atop k\} \$\$}$$

ドットの表現は連続したものを省略して表す場合に用いられる。

$$P = (x_1, x_2, \ldots, x_\ell) \qquad \texttt{\$\$ P=(x\_1, x\_2, \dots , x\_\ell)\$\$}$$

ドットには次のものが有る。

$\ldots$ \dots　$\cdots$ \cdots　$\vdots$ \vdots　$\ddots$ \ddots

総和と積分

総和 (summation) 記号は級数をあらわすときなどに使われるもので，\sum 記号で表現する。`$\sum_{i=1}^n x_i$` と書くと $\sum_{i=1}^{n} x_i$ と表示される。同じものをディスプレイモードで書くと，

$$\sum_{i=1}^{n} x_i$$

のようになる。

積分記号は，上の `sum` を `int` に置き換えた形で表現できる。つまり，$\int_a^b f(x)dx$ は `$\int_a^b f(x)dx$` として表すことができる。これも同様にディスプレイモードでは次のように表示される。

$$\int_a^b f(x)dx$$

経路積分などの丸付きの積分記号は\oint を使う。

$$\oint f(z)dz \qquad \texttt{\$\$ \textbackslash oint f(z)dz \$\$}$$

重積分は複数の\int を続ければよいが，積分記号の間が空きすぎるので，負の空白 \! を幾つか入れて調整する。

$$\int\int_{D_n} w(z)dxdy$$　　$$\int\int_{D_n}w(z)dxdy$$　負の空白なし

$$\iint_{D_n} w(z)dxdy$$　　$$\int\!\!\!\int_{D_n}w(z)dxdy$$　負の空白が 3 個

ベクトルと括弧

ベクトルは，幾何等で出てくるもので，$\overrightarrow{\rm PQ}$をあらわすのに$\overrightarrow{\rm PQ}$ と入力する。**括弧**を使って数式をバランス良く包みこむように表示することができる。そのためには，

\left ＜括弧記号＞ 数式 \right ＜括弧記号＞

という指定をする。ここで括弧記号は (, \{, [などが使えるが，このほかにも幾つか同様の方法で使える記号がある。また，片側の括弧が必要無いときは括弧記号にピリオド . を使う。

$$(1+\frac{n\pi}{\sqrt 2}\sqrt{1-\cos(\frac{\pi}{(n+2)})})^3$$　　これは\left \right を使わない

$$\left(1+\frac{n\pi}{\sqrt 2}\sqrt{1-\cos\left(\frac{\pi}{(n+2)}\right)}\right)^3$$　　これは\left \right を使っている

これらのソースは次のようになっている。

```
$$
(1+\frac{n\pi}{\sqrt 2}\sqrt{1-\cos
(\frac{\pi}{(n+2)})}\,)^3
$$
$$
\left(1+\frac{n\pi}{\sqrt 2}\sqrt{1-\cos
\left(\frac{\pi}{(n+2)}\right)}\,\right)^3
$$
```

この例で説明されていないコマンド \, がある。これは狭い空白コマンドであり微妙に空白を開けるときに使う。これらの空白を空けるコマンドは，いちいち\hspace{*n*cm} などとするよりは簡単なことから，以下のように幾つか用意されている。

空白の幅の例	コマンド	空白の幅の対比
X　　X	X\qquad␣X	\quad の倍の空白
X　X	X\quad␣X	インデントと同じ幅の空白 (英字 2 文字程度)
X X	X\␣X	適当な単語間の空白
X X	X\enskip␣X	今のフォントの英大文字の M の幅
$X\;X$	$X\;X$	\quad の 5/18(数式モードのみ)
$X\:X$	$X\:X$	\quad の 2/9(数式モードのみ)
X X	X\,X	\quad の 1/6
XX	$X\!X$	\quad の-1/6(数式モードのみ)
XX	XX	通常の 2 文字を続けたときの間隔
XX	XX	数式モードで 2 文字を続けたときの間隔

配列と行列

配列は次に示す\array コマンドで書くことができる。これは以前の\tabular 環境とまったく同様の書き方で使うことができる。

```
\begin{array}{各列の様式指定}
要素(1,1)   & 要素(1,2)   & ... 要素(1,n)   \\
...
要素(m-1,1) & 要素(m-1,2) & ... 要素(m-1,n) \\
要素(m,1)   & 要素(m,2)   & ... 要素(m,n)   ※ 最後は \\ を付けない
\end{array}
```

この array 環境を使って行列をあらわすには，前節で習った\left(\right) で挟んでやればよい。

例

$$\left(\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array} \right)$$

```
$$
\left(
\begin{array}{ccc}
a_{11} & a_{12} & a_{13}\\
a_{21} & a_{22} & a_{23}\\
a_{31} & a_{32} & a_{33}
\end{array}
\right)
$$
```

この配列コマンドを使って，**場合分け**の数式が書ける。この場合右側の括弧が無いので \right. を使い，帳尻を合わせるための透明右括弧とする。

$$h(a) := \left\{ \begin{array}{ll} a+1, & a\text{が奇数のとき} \\ a-1, & a\text{が偶数のとき} \end{array} \right.$$

```
$$
h(a) := \left\{
   \begin{array}{ll}
   a + 1, &\quad\mbox{$a$が奇数のとき}\\
   a - 1, &\quad\mbox{$a$が偶数のとき}
   \end{array}\
\right.
$$
```

番号付数式モード

数式番号の自動割付

LaTeX では数式に自動的に番号を付けてくれる環境がある。

- `equation` 環境: 1行の数式を記述
- `eqnarray` 環境: 複数行の数式を記述

このどちらの環境もディスプレイ数式モード環境なので，今まで$$で囲んで指定したのと同様の数式が番号付で表示される。

例えば，

$$E = mc^2 \tag{1}$$

は

```
\begin{equation}
E = mc^2
\end{equation}
```

と書くだけで，数式の番号を自動的に付けてくれる。また，eqnarray は何行にも渡る数式を位置をそろえて表示するときに使い，tabular や array の場合と同様に位置合わせするところに & マークを付ける。

$$\begin{aligned} |f(z)| &= |e^{z^2-4}-1|^{\frac{1}{3}} && (2)\\ &\leq (|e^{z^2-4}|+1)^{\frac{1}{3}} = (e^{Re(z^2-4)}+1)^{\frac{1}{3}} && (3)\\ &= (e^{x^2-y^2-4}+1)^{\frac{1}{3}} && (4) \end{aligned}$$

ここで，式 (**??**) は何々である。

は，

```
\begin{eqnarray}
 |f(z)| &=    &|e^{z^2-4}-1|^{\frac 13} \\
        &\leq &(|e^{z^2-4}|+1)^{\frac 13}
            =(e^{Re(z^2-4)}+1)^{\frac 13} \label{rei:second}\\
        &=    &(e^{x^2-y^2-4}+1)^{\frac 13}
\end{eqnarray}
ここで，式 (\ref{rei:second}) は何々である。
```

と入力すればよい。この例では，式中で \label コマンドを使ってラベルを付けをし，文章中で \ref コマンドを使って式番号の参照をしていることにも注意して欲しい。
※ また，この式である行だけ式番号を付けないようにするには，該当行の行末の\\ の直前に \nonumber というコマンドを入れておけばよい。
※ eqnarray 環境を使って，位置合わせだけをして，数式番号を付けない場合は eqnarray* を使うとよい。
• 自動的に付けられる数式番号は，何も指定しないと文書全体の先頭から式が出てきた順に，$1, 2, 3, \ldots$ と付いてしまう。これらを任意に指定する方法を次に説明する。

例えば，現在の節 (section) 番号と通し番号をピリオドでつないだ式番号を付けるときは，最初の式が出てくる前に次の指定をしておく。次の指定の 0 は式番号を 1 から付けるためのものなので，番号 n から付けたいときは $n-1$ と指定しておけばよい。

```
\setcounter{equation}{0}
\renewcommand{\theequation}{\thesection.\arabic{equation}}
```

直前の例を，この番号付け指定により実行すると，次のようになる。

$$
\begin{aligned}
|f(z)| &= |e^{z^2-4}-1|^{\frac{1}{3}} && (14.1)\\
&\leq (|e^{z^2-4}|+1)^{\frac{1}{3}} = (e^{Re(z^2-4)}+1)^{\frac{1}{3}} && (14.2)\\
&= (e^{x^2-y^2-4}+1)^{\frac{1}{3}} && (14.3)
\end{aligned}
$$

節 (section) 番号の代わりに自由に番号を付けたいときは，次のように指定する。syou の文字は任意であるが，既に内部で使っているものを新たに作ることはできない。また，4行目 \thesyou の \the 以下の文字と同じにする必要がある。次の指定例は，式番号を (6.30) から付ける場合についてのものである。

```
\setcounter{equation}{29}
\newcounter{syou}
\setcounter{syou}{6}
\renewcommand{\theequation}{\thesyou.\arabic{equation}}
```

数式番号を指定して表示する場合

自分で数式番号を付ける場合は，ディスプレイモードで式の後に\eqno を付けて記述する。例えば，

$$E = mc^2 \eqno{(10.2)} \tag{10.2}$$

は

```
$$
E = mc^2 \eqno{(10.2)}
$$
```

と記述すればよい。また，式番号を**左に表示させる**ときは\eqno の代わりに\leqno を使い，この場合には次のように表示される。

$$E = mc^2 \leqno{(10.2)} \tag{10.2}$$

2行の数式の中央に数式番号を付ける

二つの数式の中央に数式番号を付けるには，array 環境を用いる。なお，式番号は先に設定した番号の続きが順次付けられていくことに注意しなければならない。また，式は文中モードで表示されるので，必要に応じて\displaystyle を使い大きい式にすることができる。

例

```
\begin{equation}
\begin{array}{rcl}
 ax + by = c\\
 dx + ey = f
\end{array}
\end{equation}
```

$$\begin{array}{rcl} ax + by = c \\ dx + ey = f \end{array} \tag{14.4}$$

脚注

文章中で，ある事柄[1]に説明を付けたいときに，これをページの欄外に小さい文字で書いたものを脚注という。

例 上の文章は次のように入力されている。

> 文章中で，ある事柄\footnote{これが脚注の例である。}に説明を付けたいときに，これをページの欄外に小さい文字で書いたものを脚注という。

\footnoteにはオプションで脚注番号を\footnote[2]{なになに...}のように付けることができる。また，\footnoteはtabbular環境等の特殊な条件の中では使うことができない。そのようなところでは，番号付けと内容を別々に指定することで，脚注を付けることができる。

\footnotemark[数字]　　を表等の環境中で指定し，

\footnotetext[数字]{脚注説明文}　　を環境の外で指定する。

脚注記号を数字ではなく記号にする場合

```
\renewcommand{\thefootnote}{\fnsymbol{footnote}}
```

を\footnote や \footnotemarkを使う前に指定することで，番号の1番から9番まで順に

*,　†,　‡,　§,　¶,　‖,　**,　††,　‡‡

という記号が付く。例えば，脚注に‡が使いたいときは，上の指定に加え

```
\footnote[3]{なになに...}
または,
\footnotemark[3]
\footnotetext[3]{なになに...}
```

と指定すればよい。

[1] これが脚注の例である。

A　付録

以下の記号等は基本的に数式モードでのみ利用できる。

ギリシャ文字

読み方	小文字		大文字		読み方	小文字		大文字	
alfa	α	\alpha	A	A	xi	ξ	\xi	Ξ	\Xi
beta	β	\beta	B	B	omiclon	o	o	O	O
gamma	γ	\gamma	Γ	\Gamam	pi	π	\pi	Π	\Pi
delta	δ	\delta	Δ	\Delta		ϖ	\varpi		
epsilon	ϵ	\epsilon	E	E	rho	ρ	\rho	P	P
	ε	\varepsilon				ϱ	\varrho		
zeta	ζ	\zeta	Z	Z	sigma	σ	\sigma	Σ	\Sigma
eta	η	\eta	H	H		ς	\varsigma		
theta	θ	\theta	Θ	\Theta	tau	τ	\tau	T	T
	ϑ	\vartheta			upsilon	υ	\upsilon	Υ	\Upsilon
iota	ι	\iota	I	I	phi	ϕ	\phi	Φ	\Phi
kappa	κ	\kappa	K	K		φ	\varphi		
lambda	λ	\lambda	Λ	\Lambda	chi	χ	\chi	X	X
mu	μ	\mu	M	M	psi	ψ	\psi	Ψ	\Psi
nu	ν	\nu	N	N	omega	ω	\omega	$\Omaga$	\Omaga

アクセント

$\hat{a}$	\hat{a}	$\acute{a}$	\acute{a}	$\bar{a}$	\bar{a}	$\check{a}$	\check{a}	$\grave{a}$	\grave{a}
$\vec{a}$	\vec{a}	$\breve{a}$	\breve{a}	$\tilde{a}$	\tilde{a}	$\dot{a}$	\dot{a}	$\ddot{a}$	\ddot{a}

括弧

これらの括弧は\left や\right と組み合わせて，大きさが変化する。

(	(	)	)	[	[	]	]
{	\{	}	\}	$\langle$	\langle	$\rangle$	\rangle
$\lfloor$	\lfloor	$\rfloor$	\rfloor	$\lceil$	\lceil	$\rceil$	\rceil
/	/	\	\backslash	|	|	$\|$	\\|

関数名

arccos	\arccos	arcsin	\arcsin	arctan	\arctan	arg	\arg
cos	\cos	cosh	\cosh	cot	\cot	coth	\coth
csc	\csc	deg	\deg	det	\det	dim	\dim
exp	\exp	gcd	\gcd	hom	\hom	inf	\inf
ker	\ker	lg	\lg	lim	\lim	lim inf	\liminf
lim sup	\limsup	ln	\ln	log	\log	max	\max
min	\min	Pr	\Pr	sec	\sec	sin	\sin
sinh	\sinh	sup	\sup	tan	\tan	tanh	\tanh

2項演算子

$\pm$	\pm	$\mp$	\mp	$\setminus$	\setminus	$\cdot$	\cdot
$\times$	\times	$*$	\ast	$\star$	\star	$\diamond$	\diamond
$\circ$	\circ	$\bullet$	\bullet	$\div$	\div	$\cap$	\cap
$\cup$	\cup	$\uplus$	\uplus	$\sqcap$	\sqcap	$\sqcup$	\sqcup
$\triangleleft$	\triangleleft	$\triangleright$	\triangleright	$\lhd$	\lhd	$\rhd$	\rhd
$\bigtriangleup$	\bigtriangleup	$\bigtriangledown$	\bigtriangledown	$\unlhd$	\unlhd	$\unrhd$	\unrhd
$\vee$	\vee, \lor	$\wedge$	\wedge, \land	$\wr$	\wr	$\bigcirc$	\bigcirc
$\oplus$	\oplus	$\ominus$	\ominus	$\otimes$	\otimes	$\oslash$	\oslash
$\odot$	\odot	$\dagger$	\dagger	$\ddagger$	\ddagger	$\amalg$	\amalg

関係演算子

$\leq$	\leq, \le	$\geq$	\geq	$\prec$	\prec	$\succ$	\succ
$\preceq$	\preceq	$\succeq$	\succeq	$\ll$	\ll	$\gg$	\gg
$\subset$	\subset	$\supset$	\supset	$\subseteq$	\subseteq	$\supseteq$	\supseteq
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\in$	\in	$\ni$	\ni, \owns
$\vdash$	\vdash	$\dashv$	\dashv	$\smile$	\smile	$\frown$	\frown
$\mid$	\mid	$\parallel$	\parallel	$\sim$	\sim	$\simeq$	\simeq
$\equiv$	\equiv	$\asymp$	\asymp	$\approx$	\approx	$\cong$	\cong
$\neq$	\ne, \neq	$\doteq$	\doteq	$\propto$	\propto	$\models$	\models
$\perp$	\perp	$\bowtie$	\bowtie	$\Join$	\Join	$\notin$	\notin

大きさが変化する記号

$\sum$	\sum	$\bigcap$	\bigcap	$\bigodot$	\bigodot
$\prod$	\prod	$\bigcup$	\bigcup	$\bigotimes$	\bigotimes
$\coprod$	\coprod	$\bigsqcup$	\bigsqcup	$\bigoplus$	\bigoplus
$\int$	\int	$\bigvee$	\bigvee	$\biguplus$	\biguplus
$\oint$	\oint	$\bigwedge$	\bigwedge		

矢印

$\leftarrow$	\leftarrow	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\rightarrow$	\rightarrow	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\Updownarrow$	\Updownarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto	$\nearrow$	\nearrow
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow	$\searrow$	\searrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup	$\swarrow$	\swarrow
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown	$\nwarrow$	\nwarrow
$\rightleftharpoons$	\rightleftharpoons	$\leadsto$	\leadsto		

その他の記号

$\aleph$	\aleph	$\prime$	\prime	$\forall$	\forall	∞	\infty
$\hbar$	\hbar	$\emptyset$	\emptyset	$\exists$	\exists	$\Box$	\Box
$\imath$	\imath	∇	\nabla	$\neg$	\neg, \lnot	$\Diamond$	\Diamond
$\jmath$	\jmath	$\surd$	\surd	$\flat$	\flat	$\triangle$	\triangle
ℓ	\ell	$\top$	\top	$\natural$	\natural	$\clubsuit$	\clubsuit
$\wp$	\wp	$\bot$	\bot	$\sharp$	\sharp	$\diamondsuit$	\diamondsuit
$\Re$	\Re	$\angle$	\angle	$\backslash$	\backslash	$\heartsuit$	\heartsuit
$\Im$	\Im	$\mho$	\mho	∂	\partial	$\spadesuit$	\spadesuit

通常の文章中でも使える記号

¶ \P　† \dag　‡ \ddag　§ \S　© \copyright　£ \pounds

B　座標変換について

この節では2次元平面における座標変換について簡単な説明をしておく。2次元平面上における点$\mathrm{P}(x, y)$を反時計方向に角度θ回転して得られる点P'の座標(x', y')は以下のようにし

て表すことができる。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

この様にマトリクスとベクトルを用いた表記が便利であることから，点$\mathrm{P}(x, y)$に対してx, y方向にa, bの平行移動を行って得られる点P'の座標(x', y')も同様に表すことができる。

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

従って，次のように点$\mathrm{P}(x, y)$に対する回転を平行移動と同様に表すことができる。

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

この様な，定数である座標成分1を用いた座標の表記法を同次座標と呼ぶことにする。同次座標を用いると，(平行移動移動してから回転させるのではなく)回転してから平行移動する場合，これらの操作をまとめて表すことができる。

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & a \\ sin\theta & cos\theta & b \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

ところでここで述べた2次元における回転は，回転方向の正の向きをx軸からy軸とする際に，右ねじの進行方向をz軸に対応させた場合の3次元空間における回転としても考えることができる。この右手座標系において，点$\mathrm{P}(x, y, z)$に対してz軸を回転軸として，正の角度θだけ回転して得られる点P'の座標(x', y', z')を以下のように表すことができる。

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

同様にして，x軸およびy軸周りにおける回転もこのように表記できるが，回転の正の方向は右手座標系として一意的に決まっていくるので注意する必要がある。この右手系を鏡に映して見ることに相当するのが左手系であるが，どちらを用いるかは利用者の自由であるにしても，どちらか一つの系だけを基準にしなくてはならないのは明らかである。

余談であるが，3次元のCGでは座標変換のためにこの4次の行列を用いた積和計算を多用する。そのため，用途として3次元CGを想定している処理装置では，4×4 の行列を用いた計算を高速で行うための各種工夫をすることが一般的である。

C　TEX を使った作図

この節では，TEX を使って簡単な作図をするための方法について説明する。説明に入る前に，TEX を使った作図例を見てみることにする。先ず，横 100mm，縦 80mm の作図領域を用意し，中心の座標が (0,0) になるようにこの領域の左下角の座標を (-50,-40) と指定する。次に，この領域の境界を示す枠を描く。最後に，領域の中心から長さが 10mm,15mm,20mm,25mm,30mm,35mm,40mm,45mm の矢印を順次反時計回りに描いていくことにする。

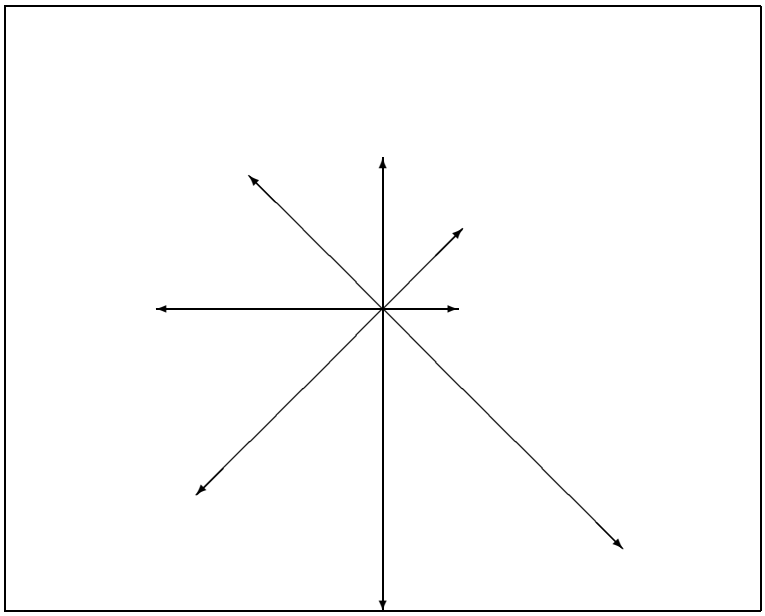


図 4: 作図例 1

この作図に用いた TEX の命令は以下のようになる。

```
%\documentstyle [12pt,hyper]{jarticle}
%\begin{document}
\begin{figure}[hbt]
\begin{center}
 \setlength{\unitlength}{1mm}
 \begin{picture}(100,80)(-50,-40)
% make frame
  \put(-50,-40){\line(1,0){100}}
  \put(-50,40){\line(1,0){100}}
  \put(-50,-40){\line(0,1){80}}
  \put(50,-40){\line(0,1){80}}
% make arrow lines
```

```
  \put(0,0){\vector(1,0){10}}
  \put(0,0){\vector(1,1){10.6}}
  \put(0,0){\vector(0,1){20}}
  \put(0,0){\vector(-1,1){17.7}}
  \put(0,0){\vector(-1,0){30}}
  \put(0,0){\vector(-1,-1){24.7}}
  \put(0,0){\vector(0,-1){40}}
  \put(0,0){\vector(1,-1){31.8}}
% 図の説明など
 \end{picture}
 \caption{作図例１}
 \label{fig01}
\end{center}
\end{figure}
%\end{document}
```

これら矢印の内，線分の長さが15mm,25mm,35mm,45mmのものは，その方向を示すベクトルのx,y成分を-1,0,1だけで表したため因子$1/\sqrt{2}$倍した値が使われている。

矢印を描く命令￥put(x,y){￥vector(v1,v2){len}}では，パラメータ(x,y)により始点，パラメータ(v1,v2)により基準ベクトルの右向きと上向き成分(ただし指定できる範囲は-4から4まで)，そしてパラメータlenにより描かれる線の長さが基準ベクトルの何倍であるかを示す。線分を描く命令￥put(x,y){￥line(v1,v2){len}}も同様であるが，方向を表すパラメータ(v1,v2)で指定できる範囲は-6から6までとなる。

どちらの命令に関しても，パラメータ(v1,v2)に対する制約のため，線分またはベクトルの角度は次の表に示す様に限定されたものとなる。

	v1=0	v1=1	v1=2	v1=3	v1=4	v1=5	v1=6
v2=0	-	0.000	0.000	0.000	0.000	0.000	0.000
v2=1	90.00	45.00	26.57	18.43	14.04	11.31	9.462
v2=2	90.00	63.43	45.00	33.96	26.57	21.80	18.43
v2=3	90.00	71.57	56.31	45.00	36.87	30.96	26.57
v2=4	90.00	75.96	63.43	53.13	45.00	38.66	33.69
v2=5	90.00	78.69	68.20	59.04	51.34	45.00	39.81
v2=6	90.00	80.54	71.57	63.43	56.31	50.19	45.00

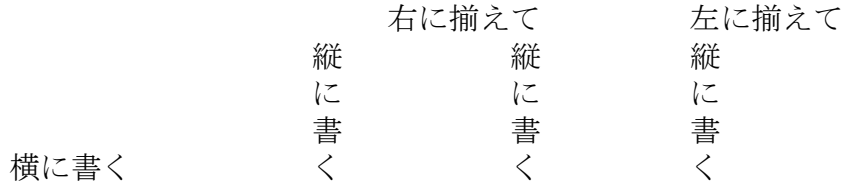
表 4: 角度(工学単位)

指定できる長さの単位にはインチやセンチメートル等の他にポイントなどがある。1ポイント(1pt)とは以下の長さのことである。

$$1\text{pt} = \frac{25.4}{72.27}\text{mm} \approx 0.35\text{mm}$$

線分の太さは標準で0.4ptだが￥thicklines命令により0.8ptになり，￥thinlines命令により0.4ptに戻すことがでる。

図中に文字を書き込むには，横書きならば￥put，縦書きならばさらに￥shortstack命令を使う。



次の図は以下の命令により書いたものである。

```
\begin{figure}[hbt]
 \setlength{\unitlength}{1mm}
 \begin{picture}(120,50)
 \put(10,0){横に書く}
 \put(50,0){\shortstack{縦\\に\\書\\く}}
 \put(60,0){\shortstack[r]{右に揃えて\\縦\\に\\書\\く}}
 \put(100,0){\shortstack[l]{左に揃えて\\縦\\に\\書\\く}}
\end{picture}
\end{figure}
```

線分を組み合わせて枠を書くこともできるが，￥rule　によっても簡単に長方形の枠を描くことができる。例えば文中において￥rule{15pt}{10pt}とすれば，横15pt縦10ptの黒い長方形 ■ が描かれる。picture環境内では，例えば￥put(30,40){￥rule{10pt}{20pt}}とすれば，点(30,40)を左下角とする長方形を描くことができる。また，例えば￥put(30,40){￥makebox(0,0){￥rule{10pt}{20pt}}}とすると，点(30,40)を重心とした長方形を描くことができる。この他，￥put(x,y){￥framebox(p,q){文字列}}とすると，座標(x,y)を左下隅として，中央に「文字列」が書かれた，縦p横qの長方形が描かれる。

幾何図形の代表の一つでもある円を描くための命令も用意されており，中心(x,y)，直径dの円を描くためには￥put(x,y){￥circle{d}}とすればよい。例えば，直径12ptの円 ◯ を描く場合には，例えば以下のような命令を用いるとよい。

```
\begin{picture}(12,12)(-2,2)
\put(6,6){\circle{12}}
\end{picture}
```

この命令では，文中に挿入して円を描くために必要な領域12pt×12ptの確保を¥picture命令で，円の位置合わせなどをパラメータ(x,y)などにより行っている。¥picture命令で描くことのできる円の直径は，最大40ポイントに制限されていることを注意しておく。また¥circle*とすれば，中まで黒い円● を描くことができるが，この場合の直径は最大15ポイントに制限される。

円と線分とを組み合わせて楕円もどきを描くこともできるが，命令¥put(x,y){¥oval(p,q)}により中心座標(x,y)，横方向p，縦方向qの長さの楕円もどきを1命令により描くことができる。例えば，横方向24pt，縦方向12ptの楕円もどき⬭ を描く場合には，以下のような命令を用いるとよい。

```
\begin{picture}(24,12)(-8,2)
\put(6,6){\oval(24,12)}
\end{picture}
```

ところで，この楕円もどきを文中に描くために，先に示した円の場合と同様のパラメータ設定を行っている。

この他，¥put(x,y){¥oval(p,q)[par]}により，par=tでは上半分，par=b,rもしくはlでは下，右もしくは左半分が，そしてpar=blなどとすれば左下四分の一が描かれる。

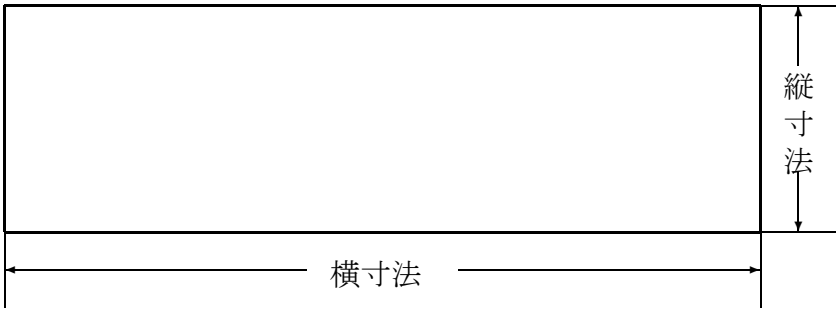
作図命令を繰り返して使うための命令¥multiputも用意されている。¥multiput(x,y)(p,q){n}{図形要素}とすることにより，「図形要素」を座標(x+kp,y+kq);k=0,1,2,///,n-1に描き連ねていくことができる。例えば，直径12ptの円5個◯◯◯◯◯ を描く場合には，以下のような命令を用いるとよい。

```
\begin{picture}(60,12)(-2,2)
\multiput(6,6)(12,0){5}{\circle{12}}
\end{picture}
```

以上説明してきたコマンド群を組み合わせることにより，例えば寸法の記入された簡単な図を描くことができる。

作図には下記のコマンドを用いている。

```
\begin{figure}[ht]
\setlength{\unitlength}{1mm}
% make frame
\begin{picture}(120,50)(-10,-10)
 \thicklines
 \put(0,0){\line(0,1){30}}
 \put(100,0){\line(0,1){30}}
```




```
 \put(0,0){\line(1,0){100}}
 \put(0,30){\line(1,0){100}}
 \thinlines
% describe dimension
 \put(0,0){\line(0,-1){10}}
 \put(100,0){\line(0,-1){10}}
 \put(40,-5){\vector(-1,0){40}}
 \put(60,-5){\vector(1,0){40}}
 \put(43,-7){\shortstack{横寸法}}
 \put(100,0){\line(1,0){10}}
 \put(100,30){\line(1,0){10}}
 \put(105,22){\vector(0,1){8}}
 \put(105,8){\vector(0,-1){8}}
 \put(103,8){\shortstack{縦\\寸\\法}}
\end{picture}
\end{figure}
```

D　レポート課題

本実験では TEX を用いた作図方法について実験を行った。理解を容易にするため，制約の多い簡単なコマンドを用いたが，この他に用意された多くのコマンドやスタイルファイル等を用いることにより，より自由度の高い作図を行うことができることを付記しておく。

さて諸君らが作成する実験レポートであるが，実験内容の簡単な説明に下記の項目につきまとめたものを加えて作成し提出してください。

(1) TEX の特徴について 400 字程度で述べなさい。

(2) 簡単な日本建築家屋の図面を作成しなさい。

(3) 3次元空間における x,y,z 軸周りの正方向の角度α, β, γによる回転をマトリックス表記し，簡単な説明を加えなさい。

以上のレポート作成には TEX を用いなさい。

参考文献

[1] 野寺 隆志，楽々 LATEX，共立出版

[2] 伊藤 和人，LATEX トータルガイド，秀和システムズ

[3] 海野 太孝，LATEX トータルリファレンス，秀和システムズ

[4] Leslie Lamport 倉沢良一監訳，文書処理システム LATEX，アスキー出版局

[5] 大野 義夫，TEX 入門，共立出版

[6] Donald E.Knuth 斎藤 信男監修，TEX ブック，アスキー出版局

[7] その他，TeX for Windows 用にいくつかの解説書が出版されている。