LaTeX 入門

1 TEX と LATEX

文書をきれいに清書するために理系の世界で広く使われているのが，TEX および LATEX と呼ばれているツールである（TEX は「テフ」または「テック」，LATEX は「ラテフ」または「ラテック」と読むこと．決して「テックス」や「ラテックス」ではない！[1]）．TEX を使うとあらゆる数式を美しく出力できるだけでなく，レイアウトや微妙な文字位置の調整，章立ての番号付けの制御など，美しい文書を作成するための多彩な機能の恩恵にあずかることができる．

現代において何か文書を作成しようというとき，最も使われているのは市販のワープロソフトであろう．ワープロは文字を入力しながら常に仕上がりの様子を画面で確認できるため，とっつきやすくて便利に思える．最近のワープロは驚くほど多機能で，非常に便利になっているのは確かである．しかし，複雑な数式など特殊な文字を多く出力したり長大な文書を作成したりするときは，まだ多大の時間と労力を消費することが多い．数学者や物理学者などの理系研究者で，こうしたワープロソフトを使って論文を書いている人は，ほとんど皆無であろう．技術的文書を作成するときに，ワープロソフトはまだまだ不便で非力なのである．

TEX や LATEX を使いこなすと，このような難点は解消される．ひとたび TEX のエキスパート（こういう人のことを「テフニシャン」という）になれば，ほとんどどんな文書でも自分の思いのままに作成できると言っても過言ではない（最終節にある「LATEX を使うとできる出力の例」を見てほしい）．今あなたが呼んでいるこの文章も，すべて LATEX を使って作成されている．最初のうちは使いにくいと感じるかもしれないが，慣れてくればこれほど便利なツールはないと必ず実感できるはずである．

TEX は元々，アメリカの著名な数学者，コンピュータ科学者である Donald E. Knuth 博士が，1970 年代に自分の著書をきれいに組版するために作り出したツールである．満足に数式を出力する手段を持っていなかった当時の理工系研究者や学生に歓迎され，TEX は広く受け入れられていった．やがて TEX をベースにより使いやすいマクロとして LATEX が Leslie Lamport 氏によって開発され，今日では TEX といえば事実上 LATEX のことを指すほどに浸透した．LATEX の登場は，少し敷居の高かった TEX をより親しみやすいものとし，技術系文書作成ツールとしてのデファクト・スタンダードの地位を確立させる契機となったと言っていいだろう．

ここではこの LATEX の最新ヴァージョンで，日本語にも対応した pLATEX 2$_\varepsilon$（ピーラテフツーイー）について，その基本的な使い方を紹介したい[2]．

---

[1] TEX は，“TEX” ではなく “E” を少し下に下げて “TEX” と書く．これは「『テックス』ではない」という意思表示でもある．また，このような書き方のできないシステム上においては，必ず “TeX” と “e” を小文字で綴る約束になっている．LATEX の場合は “LaTeX” である．

[2] 先頭についている “p” は “publishing” の略で，日本語対応であることを表している．

2 LaTeX による文書作成

2.1 基本的な流れ

ワープロソフトと LaTeX では，文書を作成していく手順が根本的に異なる．ワープロソフトにおいては，作成中の文書の状態が常に表示されている[3]が，LaTeX における文書作成の手順は，むしろプログラミングのそれに近い．すなわち，以下のような流れである：

(1) Emacs などのテキストエディタで LaTeX のソースファイルを作成・編集する．

C 言語において拡張子が ".c" のソースファイルを作るのと同じように，LaTeX では拡張子が ".tex" のファイルを作り，ここに文書内容や諸々の TeX の命令を記述していく．

(2) ソースファイルを LaTeX で処理する．

プログラミングでのコンパイルに相当する．(1) で作成したソースファイルに誤りがあると，LaTeX はエラーを出力して処理を中止する．その場合はもう一度ソースファイルに戻って，どこがおかしいかチェックしなければならない．

(3) LaTeX が出力した DVI ファイルを pxdvi などのプレビューアで表示させる．

LaTeX による処理が成功すると，いくつかのファイルが生成される．このうち，拡張子が ".dvi" のファイルが重要で，プレビューアと呼ばれるツールでこのファイルを見ると，印刷仕上がりの見本が表示される．UNIX では，xdvi というプレビューアが標準的に使われている．

(4) 表示内容をチェックして推敲し，再び (1) に戻って修正する．

出力内容には必ずと言っていいほどおかしな点が含まれているので，再びソースファイルに戻って該当する箇所を修正する．

この作業の繰り返しである．最初は面倒に感じるかもしれないが，プログラミングとほぼ同じであるから，慣れてくれば違和感はなくなってくると思う．エラーだらけだとソースファイルの修正も困難になるので，慣れないうちは少しずつ書き足してはすぐ出力をチェックしてみるとよい．

---

[3] このようなスタイルは "WYSIWYG" (*What You See Is What You Get*) と呼ばれている．

2.2 簡単な文書を作ってみよう

ここでは簡単な LaTeX のファイルを実際に作ってみることにより，LaTeX での原稿作成の流れがどのようなものかを体験してみよう．

まず LaTeX のソースファイルとして，次のようなファイルを作ってみる．Emacs を開いて下の通りに打ち込み，sample1.tex という名前で保存してみよう．

```
\documentclass{jarticle}

\begin{document}
初めて \LaTeX で文書を作りました．
\end{document}
```

次にこのソースを LaTeX で処理してみる．プロンプトで次のように入れてみよう．

```
[m23000@edu06000 ~/B]$ platex sample1.tex
```

LaTeX の処理が始まるといろいろなメッセージが表示されるが，もし先ほどのファイルが正しく書かれていれば，最後にまたプロンプトが表示されるはずである．そこで次のように打ってみよう．

```
[m23000@edu06000 ~/B]$ pxdvi sample1.dvi &
```

こうすると pxdvi が立ち上がり，実際に印刷される仕上がりの様子が確認できる．ソースファイルに打った文章が表示されていることを確認してほしい．

“?” が出て LaTeX の処理が止まってしまった人は，原稿に何らかの間違いがあるはずである．

```
! Undefined control sequence.
1.4 初めて \LaTex

?
```

上の例では，\LaTeX と打つべきところを誤って \LaTex と打ったために，定義されていない命令が呼び出されて LaTeX の処理が止まっている．この場合は x を打ってリターンすると，LaTeX は処理を中止するので，エディタに戻って該当する箇所を修正すること．

LaTeX による文書の作成作業は，基本的に以上の流れの繰り返しである．

3　LaTeX を使いこなそう

3.1　ソースファイルの構造

ここでもう少しソースファイルについて詳しく見ていく．LaTeX のソースファイルは，**プリアンブル**と原稿本体の 2 つの部分に分けられている．プリアンブルはソースの冒頭から `\begin{document}` の直前までの部分で，その原稿全体に対する変数の定義や命令を書いておくところである．

次のようなファイルを作り，`sample2.tex` として保存しよう．

```
\documentclass{jarticle}
\pagestyle{empty}

\begin{document}
\begin{center}
{\large 広島市立大学建学の基本理念（抜粋）}
\end{center}

「国際平和文化都市」を目指す広島市では平成 6 年 4 月、
科学技術の限りない発展と私たち人類の
自由な精神の悠久な発露を祈願して、
広島市立大学を開学しました。

広島市立大学では、特色のある独自の教育研究活動を通じて、
21 世紀の担い手となる人材を養成するとともに、
世界と地域が求める新しい時代の要請にも応えて、
その存在意義を深めていきたいと念じています。
\end{document}
```

これを LaTeX で処理して pxdvi でプレビューしてみよう．下のように出力されるはずである．

広島市立大学建学の基本理念（抜粋）

「国際平和文化都市」を目指す広島市では平成 6 年 4 月、科学技術の限りない発展と私たち人類の自由な精神の悠久な発露を祈願して、広島市立大学を開学しました。

広島市立大学では、特色のある独自の教育研究活動を通じて、21 世紀の担い手となる人材を養成するとともに、世界と地域が求める新しい時代の要請にも応えて、その存在意義を深めていきたいと念じています。

このサンプルを見ながら，LaTeX のソースファイルを書くにあたって気をつけるべき基本的な事柄を学んでいこう．

- LaTeX の命令はすべて，“\” で始まる．
- 日本語で書かれたところ以外はすべて半角文字で打たなければならない．LaTeX のコマンドや “\,{,}” などの文字は，全角で打ってしまうと LaTeX がエラーを起こしてしまう．
- 1 行目の `\documentclass` は，LaTeX のソースファイルの冒頭に必ず書かなければいけない命令である．この例では，`jarticle` と呼ばれる体裁で文書を作成するよう指示している[4]．
- 本文の内容は，`\begin{document}` と `\end{document}` の間に書く．`\begin{document}` の前（プリアンブル）には，この原稿での約束事などを記述する（上の例では，`\pagestyle{empty}` と書くことで，ページ番号を表示しない設定にしている）[5]．
- 半角のスペースは，いくつ続けても 1 つと同じである．例えばソースで “あい” と打ったとき，“あ い” と打ったとき，“あ　　い” と打ったときを比較してみると，2 番目と 3 番目の DVI ファイルにおける出力は全く同じ「あ い」である．
- ソースファイルで改行をしても，DVI ファイルで出力されるときは無視される．ただし，何も文字がない行で改行する（空行を入れる）と，LaTeX は段落の切れ目と判断し，改行したうえで行頭を字下げする．空行をたくさん入れても，1 つ分と同じく段落の切れ目と判断されるだけで，段落幅などは変わらない．
- LaTeX では，`#,$,%,&,_,{,},<,>,|,^,~` などの記号はコマンドの一部として使用されるので，そのまま本文中に打っても正しく出力されない．これらの文字を打つときは，`\#,\$,\%,\&,\_,\{,\}` などと打たなければならない．また `<,>,|,^,~` は，後述する数式モードで使われることになる．

大事なことは，ソースファイルで各行の右端がずれていようと余計なスペースがたくさん入っていようと，最終的な文書の仕上がりとは全く関係がないということである．文字の配置を決めるのは LaTeX の仕事であり，我々は原稿を作ることに集中すればよいのである．

---

[4] 体裁を指定するこうしたスタイルファイルには，代表的なものとしては `jarticle` のほかに `jreport`，`jbook` などがある（後述）．

[5] プリアンブルに空行を挟んでも一向に構わない．LaTeX のソースファイルで大事なのは，あくまで論理構造である．

3.2　いろいろなコマンド

ここでは，文書作成に役立つコマンドをまとめてみる．

○文字の大きさを変えるコマンド

出力例	コマンド
広島市立大学	{\tiny 広島市立大学}
広島市立大学	{\scriptsize 広島市立大学}
広島市立大学	{\footnotesize 広島市立大学}
広島市立大学	{\small 広島市立大学}
広島市立大学	{\normalsize 広島市立大学}
広島市立大学	{\large 広島市立大学}
広島市立大学	{\Large 広島市立大学}
広島市立大学	{\LARGE 広島市立大学}
広島市立大学	{\huge 広島市立大学}

上の命令は，文章中で一時的に文字の大きさを変える命令であるが，最初からデフォルトの文字の大きさを変えたい場合には，プリアンブルの 1 行目に

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

などと書く．こうするとデフォルトの文字が 12pt の大きさになる．文字の大きさは 10pt,11pt,12pt の 3 種類から選べる．何も指定しなかった場合は 10pt の文字で出力される．

○文字の種類を変えるコマンド

フォント名	出力例	コマンド
ローマン体	Hiroshima City Univ.	\textrm{Hiroshima City Univ.}
サンセリフ体	Hiroshima City Univ.	\textsf{Hiroshima City Univ.}
タイプライタ体	Hiroshima City Univ.	\texttt{Hiroshima City Univ.}
ボールド体	**Hiroshima City Univ.**	\textbf{Hiroshima City Univ.}
イタリック体	*Hiroshima City Univ.*	\textit{Hiroshima City Univ.}
小さな大文字	Hiroshima City Univ.	\textsc{Hiroshima City Univ.}
明朝体	広島市立大学	\textmc{広島市立大学}
ゴシック体	**広島市立大学**	\textgt{広島市立大学}

原則として半角文字と全角文字では上のようにコマンドを使い分ける必要があるが，\textbf だけは全角文字にも使用できる（この場合ゴシック体で出力される）．

○原稿を中央に揃えるコマンド[6]

```
\begin{center}
この文章を中央に揃える．
\end{center}
```

これは次のように出力される．

この文章を中央に揃える．

このように \begin と \end で挟まれた部分を**環境**という．上の場合は center 環境である．原稿を左揃え，右揃えにするには，center 環境の代わりに flushleft 環境，flushright 環境をそれぞれ用いる．

○箇条書き

よく使われるのは itemize 環境，enumerate 環境，description 環境の 3 つである．

```
広島市立大学は，
\begin{itemize}
\item 国際学部
\item 情報科学部
\item 芸術学部
\end{itemize}
という 3 つの学部からなる．
```

2 カ所の itemize を enumerate に書き換えると，番号付きの箇条書きになる[7]．description 環境は，見出し語と説明からなる箇条書きを出力する．

```
広島市立大学は，
\begin{description}
\item[国際学部] 入学定員 100 名
\item[情報科学部] 入学定員 200 名
\item[芸術学部] 入学定員 80 名
\end{description}
という 3 つの学部からなる．
```

各自で出力を確認してみてほしい．

---

[6]数式を中央に配置する場合は別のコマンドを用いる（後述）．

[7]箇条書きに限らず，あらゆる番号付けはこのように LaTeX に任せるべきである．自分で (1),(2) と書いていては，あとで項目を増やすことになったとき，全部直さなければいけなくなったりする．それでは LaTeX を使っている意味がない．

3.3 表の作成

技術系の文書を作成する際，データなどをまとめて提示するために表を書く機会は多い．ここでは LaTeX で表を組む方法を紹介しよう．当然ながら，ソースファイル上でデータをただ並べて書いてもうまくいかない．これまで見てきたように，LaTeX ではスペースをいくら挿入しても意味がないのである．

表を組むためには tabular 環境を用いる．下に使用例を示す．

```
\begin{tabular}{ccccc}
動物名 & 鶴 & 亀 & タコ & イカ \\
足の数 &  2 &  4 &    8 &   10
\end{tabular}
```

これは次のように出力される．

動物名	鶴	亀	タコ	イカ
足の数	2	4	8	10

最初の行にある tabular 環境の引数 “{ccccc}” はこの表の各列の体裁を指定しており，この例では表の列が 5 つであること，どの列も中央揃え (center) にするように指示している．もしここで “{crrrr}” と書いたとすると，2 列目から 5 列目までは右揃えで出力される．左揃えの場合は l を用いる．表の内容は列ごとに & で，また行ごとに \\ でそれぞれ区切らなければならない．

表の要素間に縦の罫線を引くには，tabular 環境の引数において罫線を入れたい列間に対応する箇所に | を挿入する．|| と 2 本続けて挿入すれば，縦の罫線を二重に引くことができる．一方，票に横の罫線を入れたいときは，入れたい箇所に \hline コマンドを用いればよい．これも 2 つ続けて記述すれば，横の罫線を二重に引くことができる．

次のソースと出力を参考にしてほしい．

```
\begin{tabular}{|c||r|r|r|r|}
\hline
動物名 & 鶴 & 亀 & タコ & イカ \\
\hline\hline
足の数 &  2 &  4 &    8 &   10 \\
\hline
\end{tabular}
```

動物名	鶴	亀	タコ	イカ
足の数	2	4	8	10

さて，実際に表を書くときには，表に題名と通し番号を付けたうえで,「表 1 のように……」などと引用することが多い．こうしたときに便利なのが table 環境である．これは表を貼り込むための領域を確保する環境であり，表に名前と番号を付けることができる．次のソースと出力を見てほしい．

ここで，いくつかの動物について，足の数にどのような違いがあるかを見ていこう．表 \label{tab:animal} を見てほしい．

```
\begin{table}[htbp]
\begin{center}
\begin{tabular}{|c||r|r|r|r|}
\hline
動物名 & 鶴 & 亀 & タコ & イカ \\
\hline\hline
足の数 &  2 &  4 &    8 &   10 \\
\hline
\end{tabular}
\caption{代表的な動物の足の数の比較}
\label{tab:animal}
\end{center}
\end{table}
```

ここで，いくつかの動物について，足の数にどのような違いがあるかを見ていこう．表 1 を見てほしい．

動物名	鶴	亀	タコ	イカ
足の数	2	4	8	10

表 1: 代表的な動物の足の数の比較

ここで重要なことは，表の番号を LaTeX に付けさせていることである．この表に \label{tab:animal} とラベルを貼っておき，\ref{tab:animal} と本文中でその番号を参照することで，一致した番号が出力される．自分で番号を付けてしまうと，あとで前の部分に別の表を挿入したり削除したりしたとき，本文と実際の表の番号がずれてしまい，修正に余計な時間を取られることになる．普段からこうした番号付けの方法に慣れておき，大規模な文章を作成するときにも，楽に使えるようになっておきたいものである．

なお，この番号付けを正しく出力させるには，**LaTeX で 2 回続けて処理する必要がある**．一度目の処理では番号を参照する段階でラベルの情報を読み取れていないためで，注意してほしい．

3.4 図の挿入

LaTeX で作成した文書に図を挿入したい場合がある．LaTeX はそもそも純粋に文字だけからなる文書を清書するためのシステムとして開発されたため，図の入った文書の作成にはあまり向かないとされてきた．しかし pLaTeX 2_ε では比較的手軽に、図版が組み込まれた文書を作ることができる．ここでは一番簡単な EPS 形式の画像ファイルを組み込む方法を紹介する．

LaTeX に図を挿入するためには，そのための**パッケージ**を読み込む必要がある[8]．パッケージは新たなマクロを定義するファイル群であり，これを導入するとたくさんのコマンドや環境が追加される．図の挿入には通常，graphicx パッケージを用いる．プリアンブルに以下のような 1 行を記述しよう．

```
\usepackage{graphicx}
```

これにより，LaTeX で処理する際に graphicx パッケージのコマンド群が読み込まれる．

さて，あらかじめ挿入したい画像ファイルを EPS 形式で用意しておく[9]．図を挿入したい部分に次のようなコマンドを書いてみよう．なお，ここでは中央に表示させたいので center 環境の中で使用している．

```
\begin{center}
\includegraphics{HCU-logo.eps}
\end{center}
```

エラーがなければ次のように画像が表示されるはずである．

[8] LaTeX には膨大な数のパッケージが用意されており，それらを使いこなせるようになると，ありとあらゆる文書を作成できるようになる（サンプルを参照のこと）．

[9] JPEG 形式など，一般によく使われる画像ファイルの形式は，画像を点の集まりとして捉えている．このような画像フォーマットでは，出力デバイスの解像度によっては，出力結果の美しさが損なわれてしまう．この問題を解決するために，出力の直前まで文字や図形といった要素の属性を保持するような画像形式として開発されたのが，EPS (Encapsulated PostScript) 形式である．

さらに，以下のようなオプションを付けることで，図の出力をいろいろ変更することができる．

width=w	画像の幅を指定する．height オプションで高さを指定しない場合は，画像の縦横比を変えずに拡大または縮小が行われる．
height=h	画像の高さを指定する．width オプションで高さを指定しない場合は，画像の縦横比を変えずに拡大または縮小が行われる．
scale=s	拡大率を指定する．
angle=n	回転の角度を度単位で指定する．

次のソースと出力例を参考にしてほしい．

```
\begin{center}
\includegraphics[scale=0.5,angle=90]{HCU-logo.eps}
\end{center}
```

実際にレポートなどで説明図を挿入する場合には，下の例のように figure 環境と合わせて使うとよい．これは表の作成で用いた table 環境と基本的に使い方は同じである．

```
ここで広島市立大学の校章を紹介しよう．下の図 \ref{fig:logo} を見てもらいたい．
\begin{figure}[htbp]
\begin{center}
\includegraphics{HCU-logo.eps}
\end{center}
\caption{広島市立大学校章}
\label{fig:logo}
\end{figure}
```

\caption コマンドにより，挿入した図には通し番号が付けられる．上の例では，\label{fig:logo} によってこの図に “fig:logo” というラベルを貼り，\ref{fig:logo} によってその番号を参照している．各自出力を確認してみてほしい[10]．

[10]番号を正しく出力させるには，LaTeX の処理を 2 回続けて実行する必要がある．

3.5　見出しと文書クラス

文書を作成するうえで，章や節といった階層構造を与えることは，文書全体の見通しをよくする意味でも非常に重要な作業である．LaTeX ではこうした見出しについても，単に字の大きさやフォントを変化させるといった表面的な記述ではなく，その論理構造をソースファイルに反映させることが大切である．

見出しは次のようなコマンドで記述できる．

命令	見出し
\part{*title*}	部（article グループでは使えない）
\chapter{*title*}	章（article グループでは使えない）
\section{*title*}	節
\subsection{*title*}	項
\subsubsection{*title*}	目
\paragraph{*title*}	段落
\subparagraph{*title*}	小段落

普段の文書作成でよく使用するのはおそらく \section と \subsection くらいであろう．当然のことながら，これらの番号付けはすべて LaTeX が管理してくれる．文書作成中に節を 1 つ挿入したくなっても，LaTeX で上記のコマンドを使っていれば，挿入したい箇所に書き加えるだけでよいが，直接番号付けをしているとあちこちを訂正しなければならなくなる．こういうところに LaTeX の有用性が表れる[11]．

```
\section{\TeX と \LaTeX}
\subsection{はじめに}

文書をきれいに清書するために理系の世界で広く使われているのが，
\TeX および \LaTeX と呼ばれているツールである．
```

これを処理すると次のような出力が得られる[12]．

1　TEX と LATEX

1.1　はじめに

文書をきれいに清書するために理系の世界で広く使われているのが，TEX および LATEX と呼ばれているツールである．

[11] 逆に言えば，文書の論理構造をソースファイルに反映させず，ただの清書道具として LaTeX を使うのは，ほとんど無意味な行為である．

[12] jarticle クラスでの出力結果である（後述）．

pLaTeX 2εではソースファイルを書くとき，1行目に

```
\documentclass{jarticle}
```

などと書く．この“jarticle”は，これから作成する文書がどういう類のものであるかを指定しており,「文書クラス」と呼ばれる．これによって文書全体の構造や体裁，LaTeXの処理の仕方が決定されるのである．

pLaTeX 2εで用いられる文書クラスファイルには，主に次のようなものがある．

article グループ article,jarticle,tarticle

“article”は「記事」あるいは「論説」という意味で，比較的短い文書の作成にはこのクラスを用いる．article は欧文，jarticle は横書きの和文，tarticle は縦書きの和文文書作成のための文書クラスである[13]．このクラスにおいては見出しレベルの最上位は \section であり，\chapter や \part は使えない．また，\maketitle コマンドにより文書の標題をつけると，1ページ目の冒頭にタイトルが出力され，本文が直後に従う体裁になる．

report グループ report,jreport,treport

“report”はその名の通りレポートであり，比較的長い「報告書」の類を指している．report は欧文，jreport は横書きの和文，treport は縦書きの和文文書作成のための文書クラスである．見出しは \chapter から使うことができ，\maketitle コマンドを用いた場合，標題は1ページを割いて出力される．

book グループ book,jbook,tbook

“book”はまさに本格的な書籍を意味している．book は欧文，jbook は横書きの和文，tbook は縦書きの和文文書作成のための文書クラスである．見出しは \chapter から使うことができ，必要なら \part コマンドも使用できる．また両面印刷を想定した仕様になっており，奇数ページと偶数ページでは左右のマージンやヘッダ・フッタの出力が異なっている．

この他にも，OHP シートでの仕様を想定した slides クラス，欧文用の手紙を作成するための letter クラス，アメリカ数学会のドキュメント仕様に準拠した amsart, amsreport, amsbook クラスなど，多様な文書スタイルに対応するためのクラスがたくさん作られている．

[13] article と jarticle はともに横書きだが，2つはしっかり区別して使わなければならない．欧文と和文では，文書の正しい記法がかなり異なるからである．

3.6 レポート文書などの作成

ここまで説明したこともふまえて，簡単なレポートを LaTeX で書く際の手順をまとめておこう．

○文書クラスの決定 これから作成する文書が欧文なのか和文なのか，またどの程度の規模なのか，それによって文書のスタイルも変わってくる．一般的には，和文でレポートを書く場合には，jarticle か jreport を選ぶのが適切であろう（後者を選ぶと，表紙だけで 1 ページを使うことになる）．

また，文書クラスには様々なオプションをつけることができる．すでに述べた通り，デフォルトでは基本となる文字の大きさは 10 ポイントであるが，オプションとして 11pt, 12pt を指定することにより，11 ないし 12 ポイントにすることもできる．この他にも，
○用紙の大きさを指定する（a5paper,b4paper etc.）
○用紙の方向を横置きにする（landscape）
○二段組みにする（twocolumn）
○数式番号の位置を左側に出力する（leqno）
といったオプションがある．

例えば，B5 用紙に 12 ポイントの文字の大きさで二段組みの文書を作りたいときは，

```
\documentclass[12pt,b5paper,twocolumn]{jarticle}
```

と書けばよい．

○標題 レポートなどの文書を作成する際，その文書の題目，著者名，日付などのデータが冒頭に書かれている必要がある．こうした標題情報を出力するためには，まずプリアンブルでそれらを定義したうえで，本文の冒頭（すなわち “\begin{document}” の直後）に \maketitle と書けばよい．

○ページ番号 また，ページ番号を出力するか，出力するならばどこに出力するかもプリアンブルで \pagestyle コマンドによって指定する．最も多いのは「ページ番号を出力したくない」場合で，そのときは \pagestyle{empty} と書く．ただし，empty を指定した場合でも，article グループのクラスでは標題をつけたページだけはページ番号が出力されてしまう．これを消したいときは，本文の該当箇所のどこかに \thispagestyle{empty} を書いておけばよい．

○パッケージ 文書内で使うパッケージもすべてプリアンブルの中で指定しておく．パッケージは膨大な数が提供されており，それらをうまく使えば，きわめて多様な文書にも対応することができる．図を挿入する際に使用した graphicx もパッケージの 1 つである．

○ソースをそのまま出力する方法　レポートではしばしば，C プログラムのソースなどをそのまま載せたいことがあるが，verbatim 環境を用いれば，環境内に書かれたソースがそのまま出力される．ただし，デフォルトの verbatim 環境はあまり長い文章を扱うことができないので，ある量以上のソースを載せようとすると “TeX capacity exceeded” というエラーを起こしてしまうことがある．このような場合には，verbatim パッケージを読み込んで verbatim 環境を拡張しておくことで，制限を取り払うことができる．

注意 この verbatim 環境を利用して，LaTeX のソースファイルを最初から最後まで verbatim 環境の中に入れてしまい，あたかもワープロソフトのように使おうとする人がいるが，それは**絶対に**やめてほしい．LaTeX のよいところがすべて死んでしまう．この環境は，あくまでソースをそのまま出力する必要があるときのみ用いられるべきである．

以下に具体例を示す．

```
\documentclass[12pt]{jarticle}
\usepackage{verbatim}

\title{簡単な C プログラム}
\author{齋藤　夏雄}
\date{2005 年 4 月 1 日}

\begin{document}
\maketitle
\thispagestyle{empty}

簡単な C プログラムを書いてみよう．次のソースを見てほしい．

\begin{verbatim}
#include <stdio.h>

int main(void){
  printf("Hello World!\n");
  return 0;
}
\end{verbatim}

これは画面に “Hello, World!” と表示するだけのプログラムである．
\end{document}
```

4　数式

TEX や LATEX の最大の特徴が，どんな数式も非常に美しく出力できるということである．この点に関しては市販のワープロソフトとは比較にならないと言ってもよい．LATEX が技術系の研究者や理系学生に強く支持され続けている理由の一つでもある．

4.1　数式モード

数式を書くためには，数式モードにする必要がある．通常の文書を清書するモードから数式モードに変更することにより，文字も数式にふさわしい字体になり，ギリシャ文字や数式用の各種の記号も使うことができるようになる．

数式モードにするには 2 種類の方法がある．

(1) 数式部分を `$` で挟む（**文中数式モード**）．

文章の中に数式を挿入する場合に用いる．

```
与えられた方程式 $ax-b=0$ を解くと，$x=b/a$ となる．これが
求める答えである．
```

これを LATEX で処理すると

与えられた方程式 $ax-b=0$ を解くと，$x=b/a$ となる．これが求める答えである．

となる．

(2) 数式部分を `\[` と `\]` で挟む（**ディスプレイ数式モード**）．

数式だけで独立した行を組むときに用いる．

```
与えられた方程式
\[ ax-b=0 \]
を解くと，
\[ x=b/a \]
となる．これが求める答えである．
```

これを LATEX で処理すると

与えられた方程式

$$ax - b = 0$$

を解くと，

$$x = b/a$$

となる．これが求める答えである．

となる．

注意．

a) 重要なことは，数学的な記号であれば，それが何であれ，必ず数式モードで書くということである．例えば次のような例を見てみよう．

与えられた方程式を解くと `$x=-3$` となり，求める答えは `$-3$` であることが分かる．

与えられた方程式を解くと `x=-3` となり，求める答えは `-3` であることが分かる．

このソースを LaTeX で処理すると下のようになる．

与えられた方程式を解くと $x = -3$ となり，求める答えは -3 であることが分かる．
与えられた方程式を解くと x=-3 となり，求める答えは -3 であることが分かる．

数式モードにしなかった方は，x も “x” と数式らしいイタリックになっていないうえ，マイナスの記号がただのハイフンになってしまっている．ハイフンとマイナスは全く別の記号であり，2 つ目の文はきわめて見苦しいと言わざるを得ない．このように，一見どちらでもいいと思えるような場合でも，

数学的な式や記号を記述するときは，必ず数式モードにする

ということを忘れないでほしい．

b) 前項とも関連するが，数式モード内で使う数字や記号は当然ながらすべて半角文字である．全角文字で “＄ｘ＝－３＄” などと書いても，LaTeX はそのままその文字を出力してしまう．

c) 数式モード内でのスペースは，いくつ入れても一切無視される．“`$x+1$`” と書いても “`$ x  +  1 $`” と書いても，出力結果は同じ $x+1$ である．

4.2 いろいろな数式

ここではいろいろな数式の出力方法を紹介しよう．

○累乗や添え字 累乗や添え字を持った文字を出力するには，“^” や “_” を用いる[14]．

```
$a^{2}+2ab+b^{2}=(a+b)^{2}$
$a_{n}=2a_{n-1}+3$
```

$a^{2}+2ab+b^{2}=(a+b)^{2}$

$a_{n}=2a_{n-1}+3$

○分数 分数を書くコマンドは \frac である．第一引数に分子を，第二引数に分母を指定する．

```
\[ y = \frac{1}{x+1} \]
```

これは次のように出力される．

$$y = \frac{1}{x+1}$$

一方，文中数式モードで分数を書いた場合，分子と分母が少し小さな文字で出力される．

```
$y=\frac{1}{x+1}$
```

$y=\frac{1}{x+1}$

もしこれが気に入らない場合は，$y=1/(x+1)$ と書くか，\displaystyle というコマンドを用いて $\displaystyle y=\frac{1}{x+1}$ と書けばよい．

○書体の変更 数式モード内でも書体を変更したい場合がある．このときはそれぞれ以下のような命令を用いる．

フォント名	出力例	コマンド
ローマン体	ABCabc	\mathrm{A B C a b c}
ボールド体	$\mathbf{ABCabc}$	\mathbf{A B C a b c}
サンセリフ体	ABCabc	\mathsf{A B C a b c}
イタリック体	ABCabc	\mathit{A B C a b c}
タイプライタ体	$\mathtt{ABCabc}$	\mathtt{A B C a b c}
カリグラフィック体	$\mathcal{ABC}$	\mathcal{A B C}

[14]添え字のついている文字の累乗を表現するとき，“a_{i}^{n}” とすると a_{i}^{n} と出力されるが，“${a_{i}}^{n}$” と書けば ${a_{i}}^{n}$ となって，記号の意味がより明確になる．

何も指定しなかったときはイタリック体が使われる．実際によく使うのはローマン体とボールド体程度であろう[15]．

○関数　$\sin x$ と書こうとしたとき，ただ“`$sin x$`”と書いてしまうと，“$sinx$”と出力されてしまう．イタリック体の文字は何らかの数学的対象を指していると見なされるので，これでは $s\times i\times n\times x$ ということになってしまう．このようなときは，sin が 1 つの意味を持った関数名であることをはっきりさせるために，ローマン体で書く必要がある．前項で説明したように“`$\mathrm{sin}$`”と書いてもよいのだが，sin のようによく使われる関数は，最初からそのためのコマンドが用意されているので，それを使うとよい．LaTeX で用意されている関数は次の通りである．

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

このうち，lim のように関数が添え字を伴う場合は，“_” を用いる．ただし，文中数式モードでは lim の右側に出てしまうので，分数のときと同じく `\displaystyle` を補う必要がある[16]．

```
$\lim_{n\to\infty}a_{n}=a$
$\displaystyle\lim_{n\to\infty}a_{n}=a$
```

$\lim_{n\to\infty} a_n = a$

$\displaystyle\lim_{n\to\infty} a_n = a$

[15] 線形代数学などでは，ベクトルを表すのに $\boldsymbol{x}$ というボールドのイタリック体がしばしば使用されるが，AMS パッケージが必要になるのでここでは述べない．

[16] このような仕様になっているのは，欧米では文中での数式をこのように書くのが比較的一般的だからであろう．しかし日本の数学にはあまりなじまないようだ．

○数学記号

- 演算子

出力	コマンド	出力	コマンド	出力	コマンド
$\pm$	\pm	$\cap$	\cap	$\oplus$	\oplus
$\mp$	\mp	$\cup$	\cup	$\ominus$	\ominus
$\times$	\times	$\uplus$	\uplus	$\otimes$	\otimes
$\div$	\div	$\sqcap$	\sqcap	$\oslash$	\oslash
$*$	\ast	$\sqcup$	\sqcup	$\odot$	\odot
$\star$	\star	$\vee$	\vee	$\bigcirc$	\bigcirc
$\circ$	\circ	$\wedge$	\wedge	$\dagger$	\dagger
$\bullet$	\bullet	$\setminus$	\setminus	$\ddagger$	\ddagger
$\triangle$	\bigtriangleup	$\triangleleft$	\triangleleft	$\amalg$	\amalg
$\triangledown$	\bigtriangledown	$\triangleright$	\triangleright	$\cdot$	\cdot

- 関係子

出力	コマンド	出力	コマンド	出力	コマンド
$\leq$	\le	$\geq$	\ge	$\equiv$	\equiv
$\prec$	\prec	$\succ$	\succ	$\sim$	\sim
$\preceq$	\preceq	$\succeq$	\succeq	$\simeq$	\simeq
$\ll$	\ll	$\gg$	\gg	$\asymp$	\asymp
$\subset$	\subset	$\supset$	\supset	$\approx$	\approx
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cong$	\cong
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\neq$	\neq
$\vdash$	\vdash	$\dashv$	\dashv	$\doteq$	\doteq
$\smile$	\smile	$\frown$	\frown	$\propto$	\propto
$\in$	\in	$\ni$	\ni	$\models$	\models
$\perp$	\perp	$\mid$	\mid	$\parallel$	\parallel

- 大きな関数記号

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

この表には出力が 2 つずつ書いてあるが，左側が文中数式モード，右側がディスプレイ数式モードでのものである．これらは大きさが違うだけでなく，lim と同じように添え字のつき方も異なる．もし文中においてディスプレイ数式モードと同じ添え字のつき方にしたい場合は，例によって `\displaystyle` をつける必要がある．

```
$\sum_{n=1}^{\infty}a_{n}=a$
```
$\sum_{n=1}^{\infty}a_{n}=a$

```
$\displaystyle\sum_{n=1}^{\infty}a_{n}=a$
```
$\displaystyle\sum_{n=1}^{\infty}a_{n}=a$

- 矢印類

出力	コマンド	出力	コマンド
$\leftarrow$	\gets(\leftarrow)	$\longleftarrow$	\longleftarrow
$\Leftarrow$	\Leftarrow)	$\Longleftarrow$	\Longleftarrow
$\rightarrow$	\to(\rightarrow)	$\longrightarrow$	\longrightarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown
$\uparrow$	\uparrow	$\Uparrow$	\Uparrow
$\downarrow$	\downarrow	$\Downarrow$	\Downarrow
$\updownarrow$	\updownarrow	$\Updownarrow$	\Updownarrow
$\nearrow$	\nearrow	$\searrow$	\searrow
$\swarrow$	\swarrow	$\nwarrow$	\nwarrow

- その他

出力	コマンド	出力	コマンド	出力	コマンド
$\aleph$	\aleph	∂	\partial	$\top$	\top
$\hbar$	\hbar	∞	\infty	$\bot$	\bot
$\imath$	\imath	$\prime$	\prime	$\surd$	\surd
$\jmath$	\jmath	$\emptyset$	\emptyset	$\flat$	\flat
ℓ	\ell	∇	\nabla	$\natural$	\natural
$\wp$	\wp	$\|$	|	$\sharp$	\sharp
$\Re$	\Re	$\forall$	\forall	$\clubsuit$	\clubsuit
$\Im$	\Im	$\exists$	\exists	$\diamondsuit$	\diamondsuit
$\angle$	\angle	$\neg$	\neg	$\heartsuit$	\heartsuit
$\triangle$	\triangle	$\backslash$	\backslash	$\spadesuit$	\spadesuit

○行列

行列を表示させるときには，array 環境を用いる．これは表の作成時に使用した tabular 環境と使い方は似ている．

```
$\begin{array}{cc}
a & b \\
c & d
\end{array}$
```

$$\begin{array}{cc} a & b \\ c & d \end{array}$$

実際にベクトルや行列を記述するときは通常括弧をつける．array 環境はただ行列要素を並べるだけなので，括弧は自分で補う必要があるが，そのまま（ や ）を書くだけでは，括弧が小さすぎて非常に不格好な出力となってしまう．そこで，LaTeX には \left と \right という命令が用意されており，これをそれぞれの括弧の前に書いておくと，括弧に挟まれている中身にあわせた大きさに括弧を調整してくれる．

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

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

注意しなければいけないのは，\left コマンドと \right コマンドの数は必ず等しくなっていなければならないということである．数が等しければ，区切る記号自体は違っていても構わない．左右の区切り記号の数が異なっている場合は，\right. と .（ピリオド）を使うことによって数を調整する．これを利用すると，例えば次のような出力が可能になる．

```
\[ |x| =
\left\{
   \begin{array}{rl}
   x & \mbox{if $x\ge 0$} \\
   -x & \mbox{if $x < 0$}
   \end{array}
\right. \]
```

これは次のように出力される[17]．

$$|x| = \left\{ \begin{array}{rl} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{array} \right.$$

[17] \mbox は数式内で一時的に数式モードを抜けるコマンドである．

○数式の番号付け，複数の数式の出力

数式に参照用の番号を出力したいときがある．このときは equation 環境か eqnarray 環境を用いる[18]．

```
原点を中心とする半径 $1$ の円の方程式は
\begin{equation}
x^{2} + y^{2} = 1
\end{equation}
と書くことができる．
```

これは次のように出力される．

原点を中心とする半径 1 の円の方程式は

$$x^2 + y^2 = 1 \tag{4.1}$$

と書くことができる．

さらに，複数の数式を連続して出力するときには， eqnarray 環境を用いると数式間の間隔が広く空きすぎず，自然な仕上がりになる．また，& で挟むことにより，“=” を揃えて出力する便利な機能がある．

```
\begin{eqnarray}
\int\frac{2}{4x^{2}+8x+3}dx
&=& \int\frac{2}{(2x+1)(2x+3)}dx \\
&=& \int\left(\frac{1}{2x+1}-\frac{1}{2x+3}\right)dx \\
&=& \frac{1}{2}\ln\left|\frac{2x+1}{2x+3}\right|+C
\end{eqnarray}
```

これは次のように出力される．

$$\begin{aligned}
\int \frac{2}{4x^2+8x+3}dx &= \int \frac{2}{(2x+1)(2x+3)}dx && (4.2)\\
&= \int \left(\frac{1}{2x+1}-\frac{1}{2x+3}\right)dx && (4.3)\\
&= \frac{1}{2}\ln\left|\frac{2x+1}{2x+3}\right|+C && (4.4)
\end{aligned}$$

なお，数式番号を出力せずに上のように式を揃えたい場合は，eqnarray 環境の代わりに eqnarray* 環境を用いる．

[18] array 環境では数式モードに入るために $ などで挟む必要があったが、equation 環境や eqnarray 環境は、自動的に数式モードに移行する．

○その他

• ギリシャ文字

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

ギリシャ文字は数式モード内でのみ出力可能である．上の表のように，オミクロンだけはアルファベットの o と同一なのでコマンドは用意されていない．

また，ギリシャ文字の小文字のうちいくつかについては，次に示すような異体文字が存在している．

出力	コマンド	出力	コマンド	出力	コマンド
ε	\varepsilon	ϖ	\varpi	ς	\varsigma
ϑ	\vartheta	ϱ	\varrho	φ	\varphi

特に数学で登場する ε や φ については，この異体文字の方が広く使われているので，こちらを書く方が自然だろう．

ギリシャ文字の大文字については以下の通りである．

出力	コマンド	出力	コマンド	出力	コマンド
Γ	\Gamma	Ξ	\Xi	Φ	\Phi
Δ	\Delta	Π	\Pi	Ψ	\Psi
Θ	\Theta	Σ	\Sigma	Ω	\Omega
Λ	\Lambda	Υ	\Upsilon		

上の表以外の文字は，すべてアルファベットの大文字と全く同一であるため，コマンドは用意されていない．

• 数式用アクセント記号

出力	コマンド	出力	コマンド
$\overline{AB}$	\overline{AB}	$\underline{AB}$	\underline{AB}
$\widehat{AB}$	\widehat{AB}	$\widetilde{AB}$	\widetilde{AB}
$\overbrace{AB}$	\overbrace{AB}	$\underbrace{AB}$	\underbrace{AB}
$\overrightarrow{AB}$	\overrightarrow{AB}	$\overleftarrow{AB}$	\overleftarrow{AB}

• 省略を表す点

数式の中では，省略を表す記号として“...”などを用いる．LaTeX では次のような記号が用意されている．

出力	コマンド	出力	コマンド
$\ldots$	\ldots	$\vdots$	\vdots
$\cdots$	\cdots	$\ddots$	\ddots

これらを用いると，例えば次のような行列を記述できる．

```
\[
A = \left(
\begin{array}{cccc}
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \ldots & a_{mn}
\end{array}
\right)
\]
```

$$A = \left(\begin{array}{cccc} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{array} \right)$$

\ldots と \cdots の使い分けは難しいが，カンマとカンマの間や，前後に演算子や関係子などの記号がないときは，\ldots を使い，前後に演算子などがあるときは \cdots を使うとよいだろう．

```
$a_{1},a_{2},\ldots,a_{n}$      $a_1, a_2, \ldots, a_n$
$(x-1)(x-2)\ldots(x-n)$         $(x-1)(x-2)\ldots(x-n)$

$a_{1}+a_{2}+\cdots+a_{n}$     $a_1 + a_2 + \cdots + a_n$
$x_{1}=x_{2}=\cdots=x_{n}$     $x_1 = x_2 = \cdots = x_n$
```

これ以外にも LaTeX にはいろいろなコマンドが用意されており，どんな式でも工夫すれば必ずと言っていいほど出力することができる．専門書などを読んで自分で調べ，テフニシャン目指してがんばってほしい．

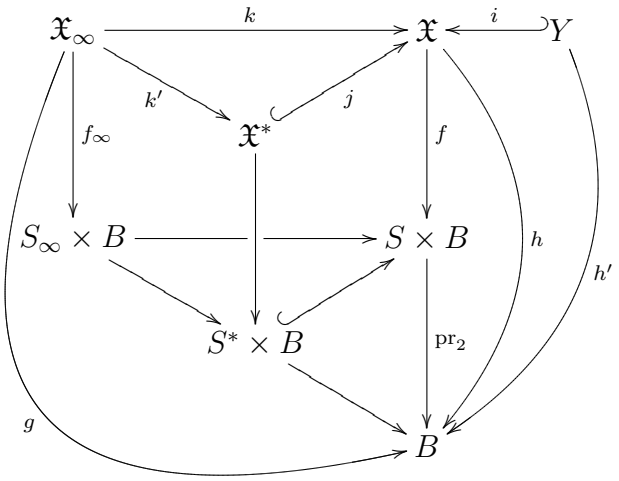
5　LaTeX を使うとできる出力の例

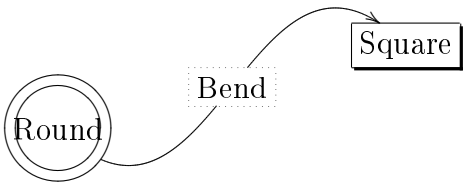
LaTeX 2$_\varepsilon$にパッケージと呼ばれるキットを追加することにより，通常の出力に加えてさらに多彩な文字や図を表現できるようになる．世界中の LaTeX の愛用者によって何百という数のパッケージが作られており，これを組み込むとさらに LaTeX は強力なツールとなる．ここでは，こうしたパッケージを組み込むことで，LaTeX がどんな文書を作れるか，その一例を紹介する．市販のワープロソフトで同じものを出力することがどれほど大変か考えてみれば，改めて LaTeX の威力を感じることができるだろう．

○数式やダイアグラム

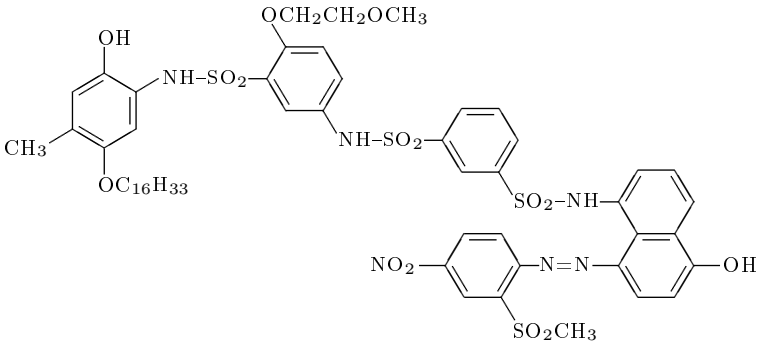
$$\frac{1}{\pi} = \frac{2\sqrt{2}}{99^2} \sum_{n=0}^{\infty} \frac{(4n)!(1103 + 26390n)}{(4^n 99^n n!)^4}$$

$$P_z^i(t) = \det(1 - F_z t, \mathscr{L}_{\bar{z}}^i) = \det(1 - F_z t, \mathrm{H}^i(\mathscr{M}_{\bar{z}}, \mathbb{Q}_\ell))$$

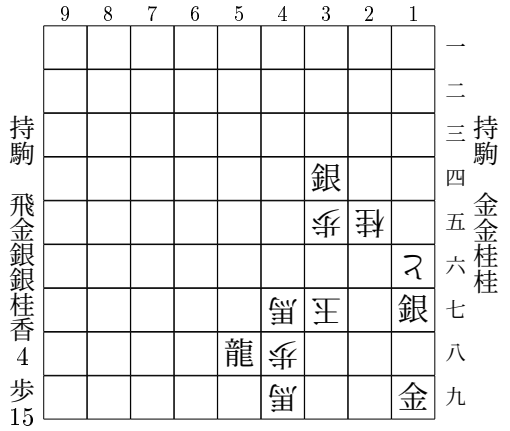




○化学式



○将棋の盤面図



詰将棋

○世界の言語

Les choses se passèrent ainsi. Nous étions cinq ou six: moi le plus vieux, leur maître, mais encore plus leur compagnon et leur ami; eux, jeunes gens à cœur chaleureux, à riante imagination... Voilà le moment, le vrai moment de réclamer de l'aide, chose d'autant plus aisée que le collègue est là, tout près, accroupi sur le dôme.

Во дни сомнений, во дни тягостных раздумий о судьбах моей родины, – ты один мне поддержка и опора, о великий, могучий, правдивый и свободный русский язык! Не будь тебя – как не впасть в отчаяние при виде всего, что совершается дома? Но нельзя верить, чтобы такой язык не был дан великому народу!

Im Jahr von der Auffrichtung vnd Beſtellung deß Großmächtigen vnnd weitläufftigen Königreichs Vtopien / 753. Als der groſſe Reichstag zu Vthen in der Haupt-Statt angangen / vnnd derowegen auß allen vmbgelegenen Landt vnnd Herrſchafften / ſo wol als auß dem gantzen Königreich / ein vnzahl Menſchen / Geiſtlich vnd Weltlich / ſich dahin verfüget hatten / beſter Hoffnung / es würde da was mercklichs außgericht werden.

○楽譜