LaTeX のマクロやスタイルファイルの利用

(Version 2.15 : 1994 年 4 月 10 日)

岩熊 哲夫[1] ・古川 徹生[2]

もくじ

[1] bear@tainsbbms, bear@hashi1.civil.tohoku.ac.jp

[2] t.furukawa.kit@tainsbbms, furukawa@hagi.ces.kyutech.ac.jp

---

この文書およびファイルの取り扱い

はじめに

この文書は，第二筆者が『LaTeX スタイルファイル Q & A 集』としてまとめていた Ver.1.0 を拡充したものです．原則として LaTeX で文書を作成したことのある方を対象としています．

ここ数年間に LaTeX に関する種々の出版物が急に揃ってきたような印象があり，このような文書は以前ほどは必要とされなくなっています．しかしながら Network News や各種掲示板では，繰り返しいわゆる FAQ (Frequently Asked Questions) に分類されるものが投稿・掲示されているのも現実です．ここでは主に東北大学内の電子掲示板を中心とした話題をもとに，マクロやスタイルファイルの利用に的を絞って文書化してみました．お仕着せのスタイルを使うためだけのスタイルファイル付き書籍のようにはならないように，マクロそのものを印字するように努め，ユーザが改変できるようにしました．ただし，出版物等からの直接の引用は，原則としてそのマクロを印字しないようにしました．

スタイルファイルは主なサーバから入手可能 (archie 等で探せそうなもの) です．このソースをコンパイルしてできた文書の脚注に「‘foo.sty’ はここにインストールされてないらしい．」と出力された ‘foo.sty’ を使いたい場合には，どうにかして手に入れる必要がありますが，この文書中で紹介したファイルのほとんどが akiu:ftp/pub/tex/latex-styles/ 以下[3]にあります．この akiu は ftp.tohoku.ac.jp のことです．ただし，学外からの場合， DNS 対応マシン以外からの anonymous ftp はできません．

紹介したスタイルファイルに対しては，第一筆者の好みに依存して力の入れ方に相当なばらつきがありますが，できるだけ使い方を示し，特に現サイトにインストールされている場合には実例を示すようにしましたので，ここで紹介したスタイルファイルを新たにインストールした後に再度コンパイルすると，巻末索引に ¶ で示した 50 個程度のスタイルファイル等についてはそのサンプルも出力されます．というわけなので，サイトのスタイルファイルの存在に依存して，この文書の出来上がり頁数が十数頁程度異なってきます．また第一筆者の環境 (すべてのスタイルファイルがインストールしてある訳ではありません) でエラーや警告が発生しないことだけを確認して作成されていますから，他の環境では Overfull 警告等が多数発生していると思います．

各サイトの TeX 自体やそのインストール状況及びスタイルファイル等の環境設定については，機種・ OS やソフト依存の部分があるなど必ずしも統一されていないので，最小限の解説に留め最後の方にまとめました．ですからこの文書では種々のマクロの使い方や例を示したかったこともあり，直接 LaTeX で作成してありますので，現状で LaTeX が問題無く利用できる状態にあるユーザーを対象としています．

和文の場合， NTT jTeX と アスキー 日本語 TeX との 2 種類の TeX[4] が存在しており，末端ユーザーにとっては困ったことも多々あります．この文書では第一筆者の都合で NTT jTeX を中心にしたような表現になっていますが，できるだけ共通に使えるような定義をする等の努力をしてあります．これについては黒木玄氏[5]に負うところ大です．また MS-DOS, UNIX 共に使えるようにしてあります[6]．本文書中の [‘latex.tex’], [‘j-art10.sty’] 等の表示は，該当マクロの定義の詳細について参照するとよいファイル名を示しています．

[3] 第一筆者の独断と偏見で集めたものは bear_collections の中．

[4] TeX は American Mathematical Society の商標．

[5] kuroki@math.tohoku.ac.jp

[6] UNIX は AT & T がライセンスしている OS． MS-DOS は Microsoft Corporation の商標．

2.1 文書サイズ設定や自分専用のマクロ等の読み込み

作成する文書が増えていくにつれて，種々のマクロを自分で定義したり，文献からコピーしたものを文書の preamble 部に置いていつも使うようになることがあります．例えば，用紙サイズ等もデフォールトより自分で設定したものの方を好むようになることもあります．こういった設定が増えてくると preamble にいつも書き込むのが煩わしくなったり，そのため逆に本文が読み難くなってしまいます．このような場合，自分自身のスタイルファイルを別ファイルとして作成し，それを常に文書に読み込むようにすると便利です．例えば，そのファイル名を ‘mydoc.sty’ とすると，文書作成の際にいつも

```
\documentstyle[11pt,mydoc]{j-article}
\begin{document}
...
```

とすると，自分の設定を読み込んだ上で本文を書くことができます．

2.2 お仕着せの環境等を変更したい

これが高じてくると，お仕着せの環境やマクロ，コマンドが気に入らなくなるかもしれません．あるいは似て非なるコマンドを定義したくなります．そうなると ‘latex.tex’ 等の定義の一部分を変更することになります．ただし，このようなファイルを直接書き変えるのではないことに注意してください．気に入らない部分を含むオリジナルのマクロを preamble あるいは自分のスタイルファイル ‘mydoc.sty’ にコピーした上で，適当な変更をすれば，この変更が有効になります．

この TEX や LATEX のコマンドを定義しているファイルは，システムの TEXINPUT(S) という環境変数に記憶されているパスにまとめてあります．これについては第 16.1 節 (p. 89) を参照のこと．また，自分用のスタイルファイル ‘mydoc.sty’ をシステムのどのディレクトリに保存しておくべきか，ということについても多少の注意が必要ですので，第 16.1 節 (p. 89) の説明も読んだ上で，実際の作業に入ってください．

この ‘latex.tex’ 等のある種のファイルは，LATEX 実行時に読み込まれる ‘lplain.fmt’ 等のフォーマットファイルに既にロードされてしまっていますから，たとえ ‘latex.tex’ の中を (間違って！) 書き変えても，結果には影響しません．一方，コンパイルの度に読み込まれる ‘article.sty’ 等の中で定義されているマクロは，このファイルを直接書き換えてしまうと，その設定が生きてしまいます．自分だけのシステムならそれでも問題は無いかもしれませんが，複数のユーザで使う場合のことや文書ファイルの互換性，元ファイルの著作権のようなものも考慮すると，直接書き換えるのは将来に問題を残すことになります．ですから，既存のマクロを自分なりのものに改訂したい場合には，必ずその部分を preamble か自分用のスタイルファイルにコピーして，そちらを改訂するようにしてください．読み込まれたファイルを順に実行していく TEX の性質上，後の設定の方が常に有効になります．

ところで，文書製造関連雑用が増えるに従って，通常の LATEX の範囲内ではできそうもないことが必要になってきます．その場合，いわゆる既存のスタイルファイルが利用できることもあります．例えば，よく出る質問のひとつに「ワードプロセサのように文章の横に枠を開けて図表を入れたい」というものがあります．これを補助するスタイルファイルのひとつに後述の ‘wrapfig.sty’ というものがあります．これを利用して文書作成をする場合には，例えば

```
\documentstyle[12pt,wrapfig,mydoc]{jarticle}
\begin{document}
...
```

とすると，枠開けコマンドが使えるようになります．この文書で紹介するスタイルファイルも，実際に使う場合にはこのようにして使います．必要なだけここに並べますが，この欄も左から順番に読み込まれますから，同じコマンドはうしろの方の定義が実際には有効になります．

2.3 @ マークが悪戯をする？

このようにオリジナルのマクロを変更したりするとすぐ気が付くように，マクロ名等に @ マークが使われています．このマークは特別な印で，通常の文書中ではこのマーク入りのマクロは使えません．ですから，例えば簡単なマクロなので preamble 部でこの @ マーク入りのコマンドを再定義して利用しようとするとコンパイルできません．逆に考えると， preamble 等文書ファイル内で定義するユーザマクロ名と LaTeX 本来のマクロ名との重複事故を避ける役目もあるのでしょう．

しかし，滅多に使わない短いマクロを変更するのに，わざわざ自分のスタイルファイルで再定義するのは面倒です．このような場合には

```
\documentstyle[11pt]{j-article}
\makeatletter                     % from here
\def\thisisapen{\@ifnextchar.......
 ..........
\makeatother                      % through here
\begin{document}
.....
```

という方法を用いることができます．この `\makeatletter` は @ マークを通常の文字として使うという宣言であり， `\makeatother` はそれを元に戻す宣言です． `\catcode`\@=11` (=`\makeatletter`), `\catcode`\@=12` (=`\makeatother`) という命令も同じ意味です．しかし特に理由が無い限り，あるいは自分が何をやっているのか明確に理解していないうちはこの方法を使わないようにしてください．特に，この文書のように頁途中で読み込むのはさらに危険ですから原則としてやめてください[7]．

ところで，フリーで手に入るスタイルファイルの中にも時折このような設定がなされているものがあり，問題を発生させることがあります．このような場合には，そのファイル中の上記の行をコメントアウトして[8]ください．例えば ‘`wrong.sty`’ がそうなってしまっていた場合，

```
\documentstyle[11pt,wrong,right,mydoc]{j-article}
```

をコンパイルすると， ‘`j-article.sty`’, ‘`j-art11.sty`’ が読み込まれた後に ‘`wrong.sty`’ が読み込まれますが，このあと @ マークが特別文字に戻されてしまいますので，次の ‘`right.sty`’, ‘`mydoc.sty`’ で用いられているかもしれない @ マーク付きマクロが正しく機能しなくなります．ただし，後述の ‘`ulem.sty`’ 等のように， @ マークがその時点で文字として使えるか否かを自動判別するマクロを用いている優れものもあります．

さらに ‘`j-article.sty`’ や ‘`j-art12.sty`’ 等の基本的なスタイルファイルを，自分のスタイルファイル中で `\input` 入力することは避け，標準の使い方通り `\documentstyle[12pt,....]{j-article}` という風に使うのを原則とすべきです．入力の順番が入れ替わると，正しい設定にならなくなる可能性があります．

---

[7] この文書では，多くのスタイルファイルを次々と入力することに起因するカウンタ不足が発生する．それを避けるために，カウンタ等を局所的に定義できるようにいい加減な変更を第一筆者が行っているが，第一筆者自身がさほど中身を理解して変更したわけではない．

[8] `\input` という手もあるが，同種ファイルの取り扱いに統一性を欠く．

2.4　文書ファイルの基本的な構成

この節は，雑用文書製造係である第一筆者のための備忘録に過ぎません．ですからこの内容は好みの問題の部分もあり，必ずしも正しい方法とは限りません．詳細については，必ずマニュアルや参考文献を読んでください．

2.4.1　始まりと終わり

LaTeX の持つ所定の書式指定の例です．

```
\documentstyle[12pt]{article}
\begin{document}
.....
\end{document}
```

です．この .... の部分に文章本体が入ります．この例は，標準文字が 12pt の大きさ (デフォールトは 10pt) の article スタイル文書です．

基本的な文書スタイルは article (簡単な文章や投稿論文等)， report (レポートや卒業論文等)， book (その名の通り，本)， letter (これも文字通り，手紙) の 4 つです．日本語の場合， NTT jTEX なら j- を， アスキー 日本語 TEX の場合は単に j をそれぞれのスタイル名の前に付けます．

2.4.2　タイトル周辺と目次

お仕着せのスタイルは， preamble でタイトル・著者名・日付をそれぞれ

```
\title{This is THE title.}
\author{It's Me!}
\date{July 4, 1993}
```

で定義しておいて， \maketitle の 1 行で出力します．

目次は通常 article 等では不要なのですが，卒論等では必須でしょう．これも単に \tableofcontents の 1 行を適当な場所に置くだけです．

2.4.3　章建て

全文をいくつかの章・節や項等で構成させる場合，例えば article では節を

```
\section{Section Title}
```

で始めます．ここにその節の表題を定義すれば，文書の初めから順に番号付きで節タイトルが表示されます．また report や book スタイルの場合は，

```
\chapter{Chapter Title}
```

から始められ，次が \section, \subsection さらに \subsubsection と続きます．その下は \paragraph と \subparagraph でおしまいです．さらに article 以外のスタイルなら， \chapter の上に \part もあり，報告書等でよく見られる『第一部』等が使えます．

2.4.4 文書本体について

文は適当な所で (できるだけ隣接する漢字同士あるいは英単語同士の間で) 改行して構いません．またパラグラフが終わったところで 空行 を入れます． \\ で強制改行するのはあまりよくありません．したがって，空行から空行までがひとつのパラグラフだと考えればよいことになります．

ワードプロセサの本来の機能ではなく空白でレイアウトしていた人が，その癖で LaTeX でも \hspace 等を乱用する場合がよくあります．しかし，まずオリジナルの機能で書いてみてください．お仕着せとは言いますが，素人の無理矢理レイアウトとは比べ物にならないくらいの出来上がりになる場合が (個人的な嗜好や遊びの場合を除けば) 多いと思います．ポスターや特別なちらし様のものでない限り，このような空白による位置合わせは避けるべきでしょう．

日本語の場合，半角文字とそうでない文字との間に半角スペースを挿入した方が見栄えがよくなります (というのは第一筆者の私的な感想です)．本当はそんなことをしなくてもいいように制御されているはずですが，面倒なので両側半角スペースにします．また \TeX 等の直後も \TeX{}␣ にします．またこのスペースは，特に NTT jTEX を使って 12pt 文書を作成する場合に Overfull 警告が出ないようにするひとつの簡単な手段でもあります．

文字の大きさは \small や \Large 等で 10 段階設定できます．またフォントも，日本語なら明朝とゴチック，英字だと **bold face** や *italic* 等の 7 種類が揃っています．異なるサイズのフォントを使う場合，フォント指定よりサイズ指定の方が先に必要です．ただ，通常の雑用文書ではそれ程多くのフォントは必要無いはずです．遊びにならないように．また罫線文字は使う必要は無いでしょう．

文字列や欧文文章あるいは \\ で長さ指定した和文文章は，それぞれ center, flushleft, flushright 環境の中に入れることによって，センタリング，左右揃えにできます．またプログラムリスト等のように，入力した文字列を一切加工せずにそのまま出力したい場合は，その部分を verbatim 環境に入れます．また文系の論文に見られるような，インデントのある引用文章は quote 環境あるいは quotation 環境中に入れます．

LaTeX の便利な機能のひとつである「相互参照」は，例えば，式表示部分で \label{eq:this} 等と定義しておけば，文中では \ref{eq:this} でその式番号を引用できるます．これがあると，式や節の並べ替えに伴う文章の変更は一切不要になるので，投稿論文等で山のように修正依頼があった時にも，めげずに頑張ることができます．この「相互参照」は，節番号・頁番号・式番号・文献番号等に使えます．

脚注は \footnote{...} の中に書きます．

2.4.5 数式を書く

数式表現そのものは，例えば $\sum$ は \sum と書く等，多少覚えて慣れる必要がありますが，その出力の美しさを考えれば，そうたいした苦労とも思われません．文中の式は \$ で囲みます．また equation 環境の式には自動的に式番号が付きます．複数行の式になる場合には， eqnarray 環境を使います．行列は数式環境下の array 環境で，表のように作成します．例えば次のような出力になります．

```
\begin{eqnarray}
F &=& \int_\Gamma \sin\,z\,dz,\qquad
    G=\sum_{n=0}^\infty a_n\,b_n \\
\mbox{\boldmath{$A$}} & \equiv &
 \left( \begin{array}{ccc}
 1+\epsilon  &  \Delta  & \zeta/2 \\
  \epsilon   & 1+\Delta & \zeta/3 \\
  \epsilon/4 & 1/\Delta & 1+\zeta \\
 \end{array} \right) \nonumber
\end{eqnarray}
```

$$
\begin{aligned}
F &= \int_\Gamma \sin\,z\,dz,\qquad G=\sum_{n=0}^\infty a_n\,b_n && (1)\\
\boldsymbol{A} &\equiv \left(\begin{array}{ccc} 1+\epsilon & \Delta & \zeta/2 \\ \epsilon & 1+\Delta & \zeta/3 \\ \epsilon/4 & 1/\Delta & 1+\zeta \end{array} \right)
\end{aligned}
$$

式番号が不要な行には `\nonumber` を付けますが，もともと式番号が必要ない `equation` 環境および `eqnarray` 環境は，それぞれ `displaymath` あるいは `eqnarray*` 環境にします．

2.4.6 箇条書き等

箇条書きにはいくつかあり，

- ひとつは，このように同種の記号で箇条に分けるもの．
- `itemize` 環境に `\item` を並べます．
- もうひとつは番号を付けるもので
  1. このようなものです．
  2. `enumerate` 環境に `\item` を並べます．
  3. さらに，それぞれに見出しを付けるものが

     見出し： このようになります．

     使い方： `description` 環境に `\item[見出し：]` を並べます．

     一般： これはすべて `list` 環境の応用で，標準で LaTeX に備わっている環境です．

2.4.7 図表

図表は，本文の流れを滞らせないように頁の上や下にまとめられるのが普通です．このように頁の中を動く「箱」をフロート (float) と呼んでいます．標準では，図と表の 2 つのフロートだけが定義されており，例えば，図なら

```
\begin{figure}[htbp]
 \vspace{3cm}
  \caption{This is the caption of this figure.}
\end{figure}
```

としておけば，キャプションの上に高さ 3cm の空白が空き，そこに図を挿入できます．図は，文章を頁に配置しながらその近辺で適切な場所に設置されますが，場所の優先順位を上の例では `[htbp]` つまり「here, その場所」，「top, 上方」，「bottom, 下方」，「page, 別頁」の順に指定していますから，その順に (LaTeX が) 最適だと思う場所に配置されます．表の場合は `table` 環境です．

表そのものは `tabular` 環境，あるいは，簡単なものなら `tabbing` 環境で作成します． 表–**1** の例は，

表 –1　これは表の例

科目名		2 年次		3 年次		4 年次	備考・合計
		I	II	I	II	I	
要望科目	応用数学	4(3)					通年講義
	物理演習		3(1)				
	応用力学			2(2)			たくさんの演習付き
	工学基礎				2(1)		仏の中の仏
	基礎工学					2(1)	鬼教官につき要注意
....							
合計		4(4)	8(7)	...	...	...	100(80)

```
\begin{table}[htb]
\caption{これは表の例}\label{tab:sample}
\begin{center}
\begin{tabular}{|l|l||c|c|c|c|c|l|} \hline
 \multicolumn{2}{|c||}{~~} & \multicolumn{2}{c|}{2 年次} &
  \multicolumn{2}{c|}{3 年次} & 4 年次 & \\ \cline{3-7}
 \multicolumn{2}{|c||}{\raisebox{10pt}[0pt][0pt]{科目名}} &
  I & II & I & II & I &
   \raisebox{10pt}[0pt][0pt]{備考・合計} \\ \hline\hline
%
 & 応用数学 & \multicolumn{2}{c|}{4(3)} &  &  &  & 通年講義  \\
 & 物理演習 &  & 3(1) &  &  &  &   \\
 & 応用力学 &  &  & 2(2) &  &  & たくさんの演習付き  \\
 \raisebox{10pt}[0pt][0pt]{\shortstack{要 \\ 望 \\ 科 \\ 目}}
 & 工学基礎 &  &  &  & 2(1) &  & 仏の中の仏  \\ \hline
 & 基礎工学 &  &  &  &  & 2(1) & 鬼教官につき要注意  \\
 \multicolumn{8}{c}{{\tt ........}} \\
 \multicolumn{2}{|c||}{合計} &
  4(4) & 8(7) & ... & ... & ... & 100(80) \\ \hline
\end{tabular}
\end{center}
\end{table}
```

の出力です．

2.4.8　参考文献リストと引用

参考文献リストは thebibliography 環境で箇条書きにします．

```
\begin{thebibliography}{99}
\bibitem{desk} Im. An. Author :
 \newblock Title of my paper,
 \newblock {\em Int. J. of Junk Papers}, Vol. 9814, No. 92,
 pp. 3002--3014, 1992.
\bibitem{chair} U. R. Ann Author and Im. An. Author :
 \newblock {\it Title of Our Book},
 \newblock We-Will Anyhow Publishing Co., Old York, 1999.
\bibitem{partition} 著者楕蔵 :
 \newblock こんな論文見たことない，
```

```
  \newblock とんでもない学会論文報告集， No. 321, pp. 3--14, 1989.
\end{thebibliography}
```

のようにしておくと，次のようなリストが現われます．

参考文献

[1] Im. An. Author : Title of my paper, *Int. J. of Junk Papers*, Vol. 9814, No. 92, pp. 3002–3014, 1992.

[2] U. R. Ann Author and Im. An. Author : *Title of Our Book*, We-Will Anyhow Publishing Co., Old York, 1999.

[3] 著者椿蔵： こんな論文見たことない， とんでもない学会論文報告集， No. 321, pp. 3–14, 1989.

上の `{99}` というのは，文献リストが二桁の数字にまで及ぶからその分のスペースが必要だ，という指示です．

これを文中で引用するには，上のリスト中 `\bibitem` のラベルを `\cite` で指定します．つまり，この場合 `\cite{chair}` が [2] のようになります．ですから，参考文献リストの順番を入れ替えても，本文引用箇所の変更は不要です．

2.4.9 索引

一度だけ「某公式集」出版を手伝ったことがありますが，最悪なことに，索引作成が仕事でした． 2 度としたくない種類の仕事です．通常は， `makeindex` という別途ソフトが必要[9]で， preamble で `\makeindex` の指示をし，本文では `\index{読み@見出し語}` で見出し語指定をします．索引そのものの作り方はマニュアルや参考文献を参照してください．

ページスタイルを設定したい

3.1 用紙のサイズを設定する

例えば，標準設定より巾を広くしたり高さを高くしたい場合があります． A4 用紙の場合には，‘`a4.sty`’ (ここにあります) と， それを広くする ‘`a4wide.sty`’ (ここにあります) とがあります． A5 の場合も ‘`a5.sty`’ (ここにあります) や， 若干プロポーションの違う ‘`a5comb.sty`’ (ここにあります) 等があります． また B4 で出力したい場合には， アスキー 日本語 TEX なら ‘`b4j.sty`’[10] というスタイルファイルを， NTT jTEX なら ‘`b4.sty`’[11] というものを使うとよいようです．以上のスタイルファイルはオプション指定するだけですから使用例は省略します．

しかし，このようなスタイルファイルがインストールされていないシステムもありますから，一番簡単なのは各自で設定することでしょう．つまり， preamble あるいは自分のスタイルファイル中で

[9] 文字列操作とソーティングなので BASIC 等でも簡単なものなら作れそうだが試してはいない．文献 [6] にもひとつの方法が示されている．

[10] ‘`b4j.sty`’ はここにインストールされてないらしい．

[11] ‘`b4.sty`’ はここにインストールされてないらしい．

```
\setlength{\topmargin}{-6mm}          % dviware dependent
\setlength{\textheight}{30cm}         % B4 size
\setlength{\textwidth}{20cm}
\setlength{\oddsidemargin}{5mm}       % dviware dependent
\setlength{\evensidemargin}{5mm}      % dviware dependent
```

などのように寸法を指定します．指定する寸法については試行錯誤をするか，参考文献にもあるように文字数や改行巾によっても設定できます．ついでに A4 の場合 (この文書の設定です) も示しておきます．紙節約のために，少し欲張った設定になっています． 1 行の文字数が多過ぎて読み難いかもしれません．

```
\setlength{\topmargin}{-11mm}
\setlength{\headsep}{20pt}
\setlength{\textheight}{24cm}         % A4 size
\setlength{\textwidth}{16cm}
\setlength{\oddsidemargin}{1.5mm}
\setlength{\evensidemargin}{1.5mm}
```

マージンは dviware 側の設定やスィッチ等によっても基準の位置が異なったり変更できたりしますから，自分のシステムに適するように試行錯誤で決めてください．この設定は NTT jTEX の j-article スタイルで，欧文のそれと同様に本文原点を紙左上端から 1 inch ずつ離れた点に，おおまかですが移動します．

横長にするのも同様で，そのサイズを指定すればよいのです．あるいは ‘mydoc.sty’ に

```
\def\landscape{%
  \@tempdima=\textwidth \textwidth=\textheight \textheight=\@tempdima}
```

というコマンドを定義しておいて，用紙サイズを指定した直後に \landscape という行を挿入してしまえば，縦横が入れ替わります．ただしマージンの微調整は必要かもしれません．

3.1.1 1 ページの行数を指定したい

正確に 1 ページの行数を一定にするのはほとんど不可能です．というのも TEX がそのようには作られてないのが原因ですが，章や節見出しや式の前後に「グルー (伸び縮みする空白)」が入っていることに加えて，文字サイズの変更によっても改行間隔が変わるからです．ただ平均的な間隔を設定する簡単な方法としては次の方法があります．

```
\def\linesparpage#1{
    \baselineskip=\textheight
    \divide\baselineskip by #1} % lineheight = textheight / line#
```

というコマンドを定義しておいて， preamble で例えば \linesparpage{20} と宣言する方法です．ただし，その文書が \maketitle を使用している場合には，その行より下で宣言しなければなりません．つまりタイトル部分のピッチは，上の方法では変更できません． \baselineskip は随所で変更されていることがあり，文書全体のピッチを変更するには \baselinestretch の方を変更します．このためには preamble で例えば

```
\renewcommand{\baselinestretch}{1.7}
```

等と宣言をします．デフォールトは 1 です．行数が指定されている講演概要集等では，この辺の微調整が必須なのですが，試行錯誤が必要で結構面倒です．

ただ単にそう宣言しただけでは，脚注の行間や図表キャプション等の行間隔も伸びる等，細かい部分にまで制御が行き届きません．それを各環境毎に多少詳細に設定するためのスタイルファイルに ‘doublespace.sty’

(ここにあります) というものもあります．オプションに指定するだけで結果は明白ですから説明は省略します．あるいはコメント欄を参照してください．

改行ピッチについては文献 [7] にもあるように，NTT jTEX と アスキー 日本語 TEX とで異なっており，前者が比較的広くなっています．第一筆者は JaWaTEX 以来の慣れのせいか個人的には前者が好きなのですが，この辺も簡単には \baselinestretch で調整する必要があります．また文字の大きさもお互いに多少違うようです．

上ではテキスト高さ固定で行間隔を変えましたが，逆に，行間隔固定でテキスト高を変えたい場合は

```
\textheight=19\baselineskip
\advance\textheight \topskip
```

のようにします [‘art10.sty’][7]．

3.1.2　1 行の文字数を指定したい

1 行の文字数を指定するのは 1 ページの行数を一定にするよりもっと難しい問題がありますが，NTT jTEX で，大まかでよければ次のようにすればできるようです．

```
\def\kcharparline#1{%
    \jintercharskip 0mm plus 0.2mm minus 0.2mm
    \settowidth{\textwidth}{あ}%
    \multiply\textwidth by #1}
```

と定義しておいて \kcharparline{40} とします[12]． アスキー 日本語 TEX 用ならば ‘genkou.sty’[13] というスタイルファイルがあるようです．これはその名の通り，原稿用紙のような出力をするための補助ファイルです．これについての例は省略します．

以上の頁の大きさや行数・文字数の設定方法は筆者の思い付きによるものを含んでおりますので，他のいろいろな方法については参考文献を参照してください．

3.2　頁毎に向きを替えたい

すべての頁を縦長・横長にするのは簡単でしたが，例えば本文は portrait の向きだが，ある表だけが landscape 向きになってしまう，といったことがあります．この場合には，‘portland.sty’ (ここにあります) というスタイルファイルが使えます (名称の由来は port[rait-]land[scape] であって地名やセメントとは関係ありません)．文中 \portrait, \landscape コマンドでの切り替えるか，portrait, landscape 環境で向きを指定します．それぞれの切り替え毎に改頁されます．使用例は省略しますが，使う前にファイル中の用紙高さ定義 \paperheight を正しく設定しておく必要があります．デフォールトは A4 サイズです．縦横の切り替えは .log ファイルに記録されていますから，それを参照してその頁だけを別に印刷する必要があります．頁毎に印刷可能なドライバ dviprt の場合は landscape 頁だけを -V オプションで出力します．頁毎に出力できるドライバが使えない場合には，第 13.11 節 (p. 79) のスタイルファイル ‘selectp.sty’ を用いて必要な頁をコンパイルすることもできますが，いずれにしても一度はコンパイルして頁を確かめる必要があります．

[12] zw という全角文字幅の単位も使えるが，JaWaTEX との整合性のために使っていない．

[13] ‘genkou.sty’ はここにインストールされてないらしい．

3.3 ヘッダ・フッタを自由に定義したい

標準的方法は \pagestyle{myheadings} を使う [5] ことでしょう．あるいは新しい頁スタイルを定義 [8] するのも比較的簡単にできます．

しかし，もっと細かい指定を簡単にしたい場合には ‘fancyheadings.sty’ を使うと良いようです． これを使うと，ヘッダ・フッタ共に 3 つのフィールドを別々に定義できます．ただし \pagestyle は \textwidth を決定したあとで宣言しなければなりません．また \headheight の値を 表 –**2-a** のよう (ただし，表は NTT jTEX のドキュメントスタイルに対する値) に設定する必要があります．この頁の例は次のような設定によるものです．時刻については第 13.2 節 (p. 76) 参照のこと．

```
\headheight=17pt \thispagestyle{fancyplain}%
\lfoot{{\small 只今 \the\hour 時 \the\minute 分}}
\rfoot{{\small ファイル名は {\tt [\jobname.tex]}}}%
\cfoot{} \chead{}%
\lhead{{\small\rm ここは \thepage 頁目}} \rhead{{\small 今日は {\tt \today}}}%
```

表 –**2** \fancyheadings 関係の設定

表 –**2-a** \headheight の設定

size	j-article	jarticle
10pt	17pt	14pt
11pt	19pt	15pt
12pt	21pt	16pt

表 –**2-b** 各フィールド

	left	center	right
header	\lhead	\chead	\rhead
footer	\lfoot	\cfoot	\rfoot

ヘッダ・フッタのフィールド名称は 表 –**2-b** のとおりですが，定義するには上の例のような宣言で

```
\lfoot{\it\today}
```

等とします． twoside スタイルを併用する場合には \rhead[EVEN-RHEAD]{ODD-RHEAD} とすれば，偶数頁と奇数頁とで異なるものを定義できます．例えば

```
\rhead[\sl\today]{\rm\thepage}
```

等とします．また節名称や項名称等をヘッダ・フッタに用いる場合には，次のようなコマンドを用いないと正しく判定ができなくなります．

```
\documentstyle[fancyheadings]{article}
\renewcommand{\sectionmark}[1]{\markboth{#1}{}}    % \leftmark
\renewcommand{\subsectionmark}[1]{\markright{#1}}  % \rightmark
  ---
\documentstyle[fancyheadings]{book}
\renewcommand{\chaptermark}[1]{\markboth{#1}{#1}}  % \leftmark
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}} % \rightmark
```

さらに \headheight, \footskip が十分な高さに指定されていれば， 2 行にわたるものも指定できます．この場合には改行を用い，

```
\rhead{\rm\thepage\\\today}
```

表 –3　罫線のデフォールト値

	line control	default	line control	default
header	\headrulewidth	0.4pt	\plainheadrulewidth	0pt
footer	\footrulewidth	0pt	\plainfootrulewidth	0pt

等とします．

ヘッダ・フッタとの間には罫線を入れられますが，その制御は線の太さの設定で行います．デフォールトは表 –3 の通りです．罫線の幅は通常は \textwidth に設定されていますが，次のような変更でマージナルノートの部分まで延ばすことができます (若干の工夫が必要だという記事の記憶がありますが，忘れました)．

```
\addtolength{\headwidth}{\marginparsep}
\addtolength{\headwidth}{\marginparwidth}
```

これで LaTeX のマニュアル [1] のような体裁になります．

表 –4　\leftmark，\rightmark のデフォールト

	book or (report + twoside)	report	article	article + twoside
\leftmark	\chapter	\chapter	\section	\section
\rightmark	\section	\chapter	\section	\subsection

LaTeX では通常第 1 頁目 (\maketitle のある頁) は plain な頁スタイルに指定されています．したがって以上の設定もこの頁には影響を及ぼさないわけで，この頁も設定したい場合には，上と異なり，

```
\pagestyle{fancyplain}
```

と宣言する必要があります．この場合の罫線の制御は前表に示した通りです．また各フィールドの定義は

```
\fancyplain{PLAIN}{NORMAL}
```

を用います．つまり， PLAIN の部分が plain の頁，つまり第 1 頁目に適応されるべき定義を示し， NORMAL の部分がその他の頁への定義です．例えば，

```
\lhead[\fancyplain{}{\bf\thepage}]{\fancyplain{}{\bf\rightmark}}
\rhead[\fancyplain{}{\bf\leftmark}]{\fancyplain{}{\bf\thepage}}
\cfoot{\fancyplain{\rm\thepage}{}}
```

とすれば，第 1 頁や章開始頁には中央フッタに頁番号が現れる以外は何も無しにできます．各フィールドのデフォールトは次のようになっています．

```
\lhead[\fancyplain{}{\sl\rightmark}]{\fancyplain{}{\sl\leftmark}}
\chead{}
\rhead[\fancyplain{}{\sl\leftmark}]{\fancyplain{}{\sl\rightmark}}
\lfoot{}
\cfoot{\rm\thepage}
\rfoot{}
```

\leftmark 等はドキュメントスタイルに依存しています (表 –4) が，\leftmark は \markboth の左で指定したもの， \rightmark はその右で指定したものか \markright で指定したものです．

3.4 頁表示に最終頁番号を入れたい

例えば報告書等で最終頁が 15 頁の時，頁表示を 3/15 等としたい場合があります．このための方法は文献 [8] にヒントがありますので，それを用いてマクロを組んでみました．ひとつの方法は \ref と \label を利用するもので，スタイルファイル中に

```
\let\thepageoriginal=\thepage
\def\thepage{\thepageoriginal/\l@stp@genumber}   % <--- this style
\def\l@stp@genumber{\@ifundefined{r@l@stp@genumber}{(last page)}%
  {\expandafter\@car\r@l@stp@genumber \@nil}}
\def\lastpagenumber{%
 \def\@currentlabel{\arabic{page}}\label{l@stp@genumber}}
```

とし， \end{document} 直前に \lastpagenumber を置くという方法です．あるいは \end{document} の定義に，上の \lastpagenumber の部分を入れてしまうこともできます．これだと 2 回で所定の表示は得られるのですが，目次と \pageref を正しくするには 3 回のコンパイルが必要になります．

この方法だと \pageref はすべて 3/15 の形式になってしまいます．頁の引用は通常のままにするには，\label も再定義して，次のような方法が考えられます．

```
\let\thepageoriginal=\thepage
\def\thepage{\thepageoriginal/\l@stp@genumber}  % <--- this style
\def\l@stp@genumber{\@ifundefined{r@l@stp@genumber}{(last page)}%
  {\expandafter\@cdr\r@l@stp@genumber\@nil}}
\def\lastpagenumber{\label{l@stp@genumber}}
\def\label#1{\@bsphack\if@filesw {\let\thepage\relax
   \def\protect{\noexpand\noexpand\noexpand}%
   \edef\@tempa{\write\@auxout{\string
      \newlabel{#1}{{\@currentlabel}{\thepageoriginal}}}}% <---- changed
   \expandafter}\@tempa
   \if@nobreak \ifvmode\nobreak\fi\fi\fi\@esphack}
```

としておいて， \end{document} 直前に \lastpagenumber を置くという方法です．これだと頁と \pageref の表示は 2 回目で正しくなりますが，目次を表示するには 3 回必要です．

しかし，以上のいずれの方法でも \thepage を再定義してしまっていますから，目次には 3/15 の形式で出てしまいます．これを避けるには，目次ファイルの作成箇所を修正する方法がありますが，ここではやはり文献 [8] の方法を用い，頁スタイルを変更する方法を示します．つまりフッタを変更してしまうものです．

```
\def\ps@totalpagestyle{%
  \let\@mkboth=\@gobbletwo
  \def\@oddhead{}\def\@evenhead{}
  \def\@oddfoot{\hfill \thepage/\pageref{lastpagenumber} \hfill}
  \def\@evenfoot{\hfill \thepage/\pageref{lastpagenumber} \hfill}}
\pagestyle{totalpagestyle}
```

をスタイルファイルで宣言しておいて，最終頁 \end{document} 直前に \label{lastpagenumber} を置くだけです．あるいは \end{document} の中にこのラベル定義を含めて再定義してしまっても同じことです．これだと 2 回で所定の表示になります．ただしタイトル頁を持つ場合には，第 4.1 節 (p. 18) で示すように \maketitle のすぐ次に続けて \thispagestyle{totalpagestyle} を付ける必要があります．この頁の例は最後の方法を用いてあります． この最後の方法は ‘nofm.sty’[14] で用いられています．これは article を対象としていますが，オプションに設定するだけで頁表示が変わりますから例は省略します．

[14] ‘nofm.sty’ はここにインストールされてないらしい．

3.5　8pt 等の文書を作成したい

通常は 10pt がデフォールトで， 11, 12pt の設定しかありません．版下原稿提出という場合でも，たいていは最終的な製本段階で縮小されますので，通常サイズで間に合うことが多いのですが，時には最終原稿そっくりの 8pt 文字サイズ文書等の出力が必要になるかもしれません．これをするスタイルファイルのうち，欧文用なら ‘xarticle.sty’[15] を用いればよいようです．このスタイルファイルの元でなら 7pt, 8pt と 9pt までの文書が作成できます．ただし ‘art7[8,9].sty’ が同時に必要です．一行目を

```
\documentstyle[8pt]{xarticle}
```

とするだけですから例は省略します．和文用のはあるのかしらん．

3.6　行番号を付けたい

草稿段階等で行番号を付けたいことがあるかもしれません． ‘numline.sty’ (ここにあります)　を用いると， 5 行毎に番号を付けてくれます．頁毎に 1 から始めることも可能です．オプションに入れれば明らかですから例は省略します．

3.7　欧文文書で右揃えをやめたい

タイプライタで作成したように，右揃えをやめて文書を作成することもあるかもしれません．しかし元々の \raggedright ではハイフンを極力避けるようになっており，狭い幅の文書では問題があります．これを多少改善したものに ‘raggedri.sty’ があります． \RaggedRight コマンドと FlushLeft 環境が使え，ハイフンを使った右揃えのない文書が作成できます． \RaggedRight と宣言すればいいだけですし，環境 FlushLeft は flushleft 環境と同様に使えます．次の例は，左側が通常の flushleft 環境で，右側が FlushLeft 環境になります．

This example shows difference between two environments, **flushleft** and **FlushLeft**. This paragraph must be flush-lefted. Both paragraphs are set in the minipages with the same width.

This example shows difference between two environments, **flushleft** and **FlushLeft**. This paragraph must be flush-lefted. Both paragraphs are set in the minipages with the same width.

オリジナルファイルにある例題で，スタイルファイル名を raggedri としたり raggedright としたりしていることに注意してください．実際に得られるファイル名は前者のようです．

3.8　草稿作成段階にはラベルを忘れたくない

草稿の段階で，式や文献の引用をする場合のラベル等を覚えておく必要がありますが，文章が長くなると大変です[16]．したがって，草稿の段階では \label や \cite で指定したラベルを欄外や行間に示しておくと便利です　．このためのスタイルファイルが ‘showkeys.sty’ です．単にオプションに入れてコンパイルすれば結果は明らかですが，例えば \cite{total} はこのように $\overset{\texttt{total}}{[6]}$ なります． \label{eq:sk} の定義部は

$$a_n = \frac{1}{\ell}\int_0^{\ell} f(x)\,\exp(-in\omega x)\,dx \qquad (2) \quad \boxed{\texttt{eq:sk}}$$

15 ‘xarticle.sty’ はここにインストールされてないらしい．

16 同様の機能を有する ‘draft.sty’ というのもあるが，これを長い文書で用いると ‘too many unprocessed float’ という警告が発生することがある． draft は通常の LaTeX で既に予約されたスタイルなので，このファイルは \input するか改名して使う．

となり，その引用部 \ref{eq:sk} は式 eq:sk (2) のようになります．式や章建てでの定義部はこのように欄外に出るからいいのですが，本文中に \label を使うと文章と重なるので注意してください．

3.9 タイプライタのように所定の位置に文字を打ち込む

これは厄介です．たいへんな手間がかかります．科学研究費用の書式ファイル等もあるようですが，ともかく大変です．手書きの習慣が相変わらず残っているために，枠の中に書き込まなければならない書類が無くなる気配は一向にありません．が，これを補助してくれそうなものがありましたので紹介します．ただし使ってみてはいません．ファイルは ‘fillform.tex’ (ここにあります) です．これは picture 環境の枠を使って所定の場所に所定の大きさの箱を空けるのですが，その場所決めに会話形式を採用したものです．予め書式の中の枠の位置と大きさとを測定しておけば，この「プログラム」で書式の決まった文書の印刷が可能になります．会話形式で入力する方法に関する例題としても面白いものです．コンパイルしてみてください．

もうひとつのスタイルファイルは ‘at.sty’ (ここにあります) です．これも同様のことをしますが，会話形式にはなっていません．例えば原点から右に 2cm，下に 5cm なら

```
\at(2cm,5cm){\parbox[t]{10cm}{this is a box}}
```

となるようです．ただし紙の左上隅から原点までの 1 inch の設定をデフォールトとしていますから，ドライバのスィッチによってはその分ずれる可能性があるので注意してください．

3.10 手分けして書いた文書を合体させるには

ある文書を手分けして書いたり，あるいは最近の自然科学の本にあるように，異なる執筆者によるそれぞれが独立した文章を連ねたものを作成することは簡単でしょうか．もちろんそれぞれのファイル作成時に必要な \documentstyle 等のコントロール行は省いた後で合体させます．

文書ファイルを入力するコマンドには \input と \include のふたつがあります．前者は単にファイルそのものを合体させたのと同じ結果になります．

後者の場合は読み込むことになっているファイルの指示と，実際に読み込むべきファイルの指示の 2 つが必要になります．最終的に読み込むべきファイルはすべて \include でその順に文書ファイルに指定します．そうした上で preamble 部に \includeonly で実際にコンパイルすべきファイル名を指示します． \input との違いは，各ファイルを入力した箇所で改頁が行われること．さらに，既にある .aux ファイルの情報を順に入力した上で指示されたファイルだけをコンパイルしますから，部分的なコンパイルにもかかわらず，頁番号等は完成時のそれと一致するようになります．目的に応じて使い分けてください．

article スタイル文書をまとめて report スタイルにするのは簡単ですが，ここでは article スタイルでそれぞれ書かれた論文をまとめて，やはり article スタイルの学科紀要のようなものを作成するものをひとつ作ってみました． ‘cllctdpp.sty’[17] がそれですが，中身は相変わらず行き当たりばったりのマクロ群です．試してみたことは，

- 目次に題目だけでなく，各論文の著者も載せた．
- 二段組だが各論文の最終頁は段をバランスさせて，その下に概要を一段で置いた．
- 当然だが，論文毎にカウンタはリセットされる．

[17] ‘cllctdpp.sty’ はここにインストールされてないらしい．

- fancyheadings.sty を用いて，各論文第一頁ヘッダには，その論文の最初と最後の頁番号が付く．
- コンマで仕切られた著者名を加工してみた．第 15.5 節 (p. 88) 参照．
- 各論文同士の label の重複を避けられる．

といったことです．いずれも他力本願です．

3.11 窓の空いた封筒用の手紙は書けませんか

よく請求書等にあるように，手紙に書いた宛先が窓の空いた封筒から見えるようにできないでしょうか．どうやら 'meinbrf.sty'[18] がそれ用に使えるようです．宛先や差しだし人住所を別ファイルにしておける点も便利かもしれません． あるいは 'brief.sty'[19] も同じ目的かもしれませんが，マニュアルの言語がオランダ語かな? よく分かりません．英語やドイツ語にも変換できるようにはなっているようです．

3.12 類似の手紙を複数箇所へ

同じ様な内容の手紙を複数の相手に出す必要が，雑用が増えれば増えるほど多くなります．それを援助してくれるのが 'formlett.sty' です．マニュアルや例題が揃っていますから，ここで解説の必要は無いと思いますが，一番簡単な使い方を示しておきます．例は，ふたりの学生の実習を受け入れてくれたふたつの会社へのお礼状です．

```
\input formlett.sty

\initlabels[12pt]{j-article}{\setlength{\topmargin}{0mm}%
\setlength{\headsep}{20pt}%
\setlength{\textheight}{23cm}%
\setlength{\textwidth}{15cm}%
\setlength{\oddsidemargin}{6.5mm}%
\setlength{\evensidemargin}{6.5mm}%
}

\beginletter

\loaddefaultparas % <== デフォールトを使うために必要

\NOPAGENUMBERS
\mbox{}\hfill\paras[1]\bigskip

\noindent\paras[3]\medskip

\noindent{\large\bf\paras[4]}\bigskip

\noindent 拝啓　時下益々ご清祥のこととお慶び申し上げます．\medskip

さて，この夏は当学科 3 年次学生の\paras[1][2] の実習につきまして，
貴社の\paras[2][2] での\paras[3][2] に関する業務補助と．．．
\medskip
```

[18] 'meinbrf.sty' はここにインストールされてないらしい．
[19] 'brief.sty' はここにインストールされてないらしい．

```
何卒，今後ともよろしくお願いいたします．

\mbox{}\hfill 敬具

\mbox{}\hfill
 \figbox[4cm]{○△大学土木工学科\par\noindent 何野某　主任教授}
\vfill\eject
\endletter

\defaultmarks
\paradefaults%
 \today;; 総務部総務課; 総務課長殿%
+ 内野楽聖; 工事事務所; 測量!

\blockmarks\beginblockmode
........ 次の空行を活かすために必要 <==

〒 160
 \addressbox[6.5cm][1em]{東京都新宿区四谷一丁目% <==
   無番地\par\noindent 東京建設四ッ谷事務所}
 山田太郎様
--------------
 鈴木次郎
 新宿工事事務所

==============
........
 平成 5 年 11 月 1 日

 \addressbox[6.5cm][1em]{東京都新宿区四谷一丁目% <==
   無番地\par\noindent 土木コンサル総務課}
 山本三郎総務課長殿
--------------
 鈴木四郎
 代々木設計事務所
 擁壁設計
==============

\endblockmode
\firstread{\global\textwidth=17cm}
\labelsquit[][\large\raggedright][2][][6.5cm]{}
\end{document}
```

この例のように，宛先だけでなく手紙文中の任意の語句にも，手紙ごとの設定が可能ですから便利です．すべてのパラメータにデフォールトが設定できます．また最後の郵便ラベル出力も，通常の `letter` スタイルと異なり二段で上から順に出せますから，手差しする剥離紙の有効利用が可能です．他にもいくつかパラメータの指定方法があります．ドキュメントや例題を見て，使い易いものを選択してください．

3.13 マニュアルを作りたい

自分で作成したプログラムやライブラリのマニュアルを UNIX の `troff` による `man` 出力に類似した形式で作成するのに適したスタイルファイルが ‘`manpage.sty`’ (ここにあります) です．スタイルオプションに指定するだけですし，例題が付いていますので，説明は省略します．

3.14 ある出版社刊行論文のスタイルファイル

Elsevier では自社の科学論文集への投稿用のスタイルファイルを配布しています．'espart.sty' (ここにあります) がそれですが，投稿したい論文集の如何によらず，このスタイルファイルで作成された文書は，出版が決定した段階で Elsevier の方で処理され，それぞれの論文集のレイアウトに変換されます．これもマニュアルが付属していますのでここでは説明を省略しますが，最終的なスタイルファイルは公開せずに版下原稿ファイルを提出させるという訳です．他に数学物理に関するマクロが数種定義されています．

3.15 その他のスタイルの文書作成

需要に沿って種々のスタイルがあるようです．いくつか挙げますが，例や解説は割愛します．問題に対するヒントや解答を順次入力しておいても，最終的には問題の頁やヒント・解答の頁とを振り分けて文書作成をしてくれるのが 'answers.sty'[20] や，'ans.sty'[21] です．あるいは 'exercise.sty'[22] も同様です．少し違いますが 'exam.sty'[23] は試験問題用のスタイルです．

他に，料理のレセピ用のスタイルファイル 'recipe.sty'[24] や，履歴書作成用の 'resume.sty'[25] ，あるいは 'vita.sty'[26] 等があります． 'schedule.sty'[27] は簡単なスケジュール表を作成するためのものです．

タイトル周辺について

4.1 empty スタイルなのにタイトル頁だけ頁が出てしまう

これは \maketitle のマクロの中に \thispagestyle{plain} という行があるからです．これを無視させるには次のようにします (comp.text.tex からの引用)．

```
\maketitle\thispagestyle{empty}   % No Line Break
```

このように \thispagestyle を \maketitle にすぐに続ければいいようです [8]．

4.2 もう少し簡単なタイトル表示にしたい

自分だけのメモ等に author というのも物々しいので，単にタイトルと日付だけにして，しかも少ない行数に表示したい場合があります．これには例えば次のようなマクロで対処できます．

```
\def\makesimpletitle{%
 \def\@maketitle{\newpage\null\vskip .5em%
  \@ifundefined{@author}{{\LARGE \@title}\hfill(\@date)\par\vskip .5em}%
  {\begin{center} {\LARGE \@title} \end{center}%
                \begin{flushright} \@author~(\@date) \end{flushright}}%
```

[20] 'answers.sty' はここにインストールされてないらしい．

[21] 'ans.sty' はここにインストールされてないらしい．

[22] 'exercise.sty' はここにインストールされてないらしい．

[23] 'exam.sty' はここにインストールされてないらしい．

[24] 'recipe.sty' はここにインストールされてないらしい．

[25] 'resume.sty' はここにインストールされてないらしい．

[26] 'vita.sty' はここにインストールされてないらしい．

[27] 'schedule.sty' はここにインストールされてないらしい．

```
  \par\vskip 1em}%
 \maketitle \let\makesimpletitle\relax}
```

これで \maketitle の代わりに \makesimpletitle とすればいいだけです．\author{...} を定義した場合にも，デフォールトよりは簡略なレイアウトになります．二段組にも対応しています．

4.3 二段組文書で概要だけは一段で出力したい

通常二段組に abstract 環境で概要を記述すると，タイトル部分が終わったすぐ次の左段最初に概要が位置してしまいます．\maketitle を使わないで \twocolumn[title ... abstract ....] ... を用いて手動でレイアウトする方法が一番簡単でしょうが，もうひとつの方法としては，次のようなスタイルが考えられます．また quotation 環境を使わずに \parbox を使えば，好きな幅で概要を書くことができます．その場合には center 環境の中に \parbox を入れるのを忘れないようにしてください．

```
\def\abstract#1{\long\def\@abstract{#1}}
\def\@abstract{}

\let\@oldmaketitle=\@maketitle
\def\@maketitle{
    \@oldmaketitle
    \begin{center}\large\bf Abstract\end{center}
    \begin{quotation}\@abstract\end{quotation}
    \vskip 1.5em}
```

としておいて，使用する場合には文書の最初の方で

```
\title{This is THE title of my paper}
\author{It's ME!}
\abstract{We've discovered...}
\maketitle
```

とするだけです [‘article.sty’]．もし \def\abstract の中の \long を除くと，概要の文章中で複数のパラグラフ，つまり空行による改行ができません．通常の論文等ではむしろそのような長い概要になることはないので，この \long は必要が無いかもしれません．上の方法は単純なものなので，もう少し複雑な場合に対して書き直したものの例を ‘1-in-2.sty’ (ここにあります) に定義してみました．サンプルも付いていますのでそちらをご覧ください．

4.4 二段組で \thanks の脚注を一段にしたい

もちろんこれは第 7.3.1 節 (p. 28) の ‘multicol.sty’ で簡単に対処できますが，通常の twocolumn 文書でもごまかせます．それには，第一頁の左段に文書全幅で脚注を出力し，右段にはそれと同じ高さの脚注を空白だけで出力する方法を用います．できるだけ自動化したいので， \par の定義を一時的に変更する方法を用いた例を第 4.3 節 (p. 19) の ‘1-in-2.sty’ (ここにあります) で定義してみました．

4.5 もっと大幅にタイトルのスタイルを変更したい

これには \@maketitle を再定義するのが適当です．要領は上の \abstract の追加と同じですが，ここでは例えば日付の表示をやめ，英語タイトルや所属等を追加する場合を説明します．特に国際会議概要のように，タイトル部分をある高さの中にレイアウトするような場合を想定してみました．

論文のタイトルがここにくる

Here is the title in English

名無しの権兵衛

北東大学工学部木土工学科

この論文の要旨はここに書かれることになるため，本文が二段組
であっても，この部分だけは一段組で組まれる

Key Words : *twocolumn, maketitle, abstract ...*

1. はじめに

ここからが本文．二段組の場合は，これが左側の段になる．...

二段組ではここが右側の段として現われるはずだが...

図 –**1** 大幅にタイトル部分を変更した例

```
\newdimen\vspace@title  \newdimen\vspace@abstract
\vspace@title=0.15\textheight              % space for title
\vspace@abstract=0.15\textheight           % space for abstract
%
\def\etitle#1{\gdef\@etitle{#1}}           % english title
\def\affiliation#1{\gdef\@affiliation{#1}} % affiliation
\def\abstract#1{\gdef\@abstract{#1}}       % does NOT allow paragraphs
\def\keywords#1{\gdef\@keywords{#1}}       % keywords
%
% minor-font definition for {'KEY WORDS:'}
\font\fontkeywords=cmbxti10                 % bold italic
\def\keywordsname{\fontkeywords Key Words :}
%
\def\maketitle{\par
 \begingroup
   \def\thefootnote{\fnsymbol{footnote}}%
   \def\@makefnmark{\hbox
       to\z@{$\m@th^{\@thefnmark}$\hss}}%
   \if@twocolumn
     \twocolumn[\@maketitle]%
     \else \newpage
     \global\@topnum\z@
     \@maketitle \fi\thispagestyle{plain}\@thanks
 \endgroup
 \setcounter{footnote}{0}%
 \let\maketitle\relax
```

```
  \let\@maketitle\relax
  \gdef\@abstract{}\gdef\@keywords{}%           to save memory ???
  \let\abstract\relax\let\keywords\relax%
  \let\etitle\relax\gdef\@etitle{}%
  \let\affiliation\relax\gdef\@affiliation{}%
  \gdef\@thanks{}\gdef\@author{}\gdef\@title{}\let\thanks\relax}
%
\def\@maketitle{\newpage\null
%                                                                    title
\vbox to \vspace@title{%
 \begin{center} {\LARGE \@title} \end{center} \vfill
 \begin{center} {\large \@etitle} \end{center} \vfill
%                                                                   author
 \begin{center} {\Large \lineskip .5em \@author} \\
    \@affiliation \end{center} \par} \vskip .5ex
%                                                                 abstract
\vbox to \vspace@abstract{\vfill%
 \@ifundefined{@abstract}{}{%
 \begin{list}{}{\setlength{\leftmargin}{0.15\textwidth}
  \topsep=0pt \partopsep=0pt \listparindent=1em \parsep=0pt
  \setlength{\rightmargin}{\leftmargin}
  \setlength{\baselineskip}{0.8\normalbaselineskip}}
 \item {\small \@abstract}
%                                                                 keywords
 \@ifundefined{@keywords}{}{%
  \begin{list}{\keywordsname}{
   \topsep=-0.8\parskip \partopsep=0pt
   \settowidth{\labelwidth}{\keywordsname}
   \settowidth{\leftmargin}{1.1\keywordsname}
   \setlength{\baselineskip}{0.8\normalbaselineskip}}
  \item {\small\it\@keywords}
  \end{list}}
 \end{list}}
 \par \vfill}}
```

としておいて，文書の最初の方で各種定義をすればいいのです．

```
\title{This is THE title.}
\etitle{Title in English}
\author{It's ME!}
\affiliation{Dept. Goofy Eng. Tonpei University}
\abstract{Blah blah blah.....}
\keywords{Lazy bone, Goofy, ...}
\begin{document}
\maketitle
....
```

上の \maketitle の再定義は原則としては必要ないのですが，ここでは，新たに定義した \@etitle や \keywords 等を初期化してしまうことによって，メモリーの節約をしています (本当はどの程度効き目があるのか知りません)．上の例は二段組の本文で用いても，タイトルや概要までは一段で表示されます． 図–**1** がその出力例です．概要集や予稿集等では，『タイトルは○行分，概要は△行分に・・・』といった指定があります (ワードプロセサ全盛になってからはほとんど守られていないようです) が，この高さを \vspace@title, \vspace@abstract で指示します．その他のレイアウトについては簡単に修正できるでしょうから，説明は省略します．この方法は土木学会論文集用のスタイルファイル[28]を作成した時に用いたものです．

[28] akiu:ftp/pub/tex/latex-styles/jsce/ に 'jsce.sty' 等を例題・マニュアルとまとめて置いてある．

目次に関して

5.1 目次の深さを変えたい

tocdepth というカウンタが目次の深さを決めており，この値が大きいほど細かいセクションまで目次になります．値は右の表のようになっているので，もしsubsubsection までを目次に入れる場合には

```
\setcounter{tocdepth}{3}
```

とすればよいのです [7]．

0	:	chapter
1	:	section
2	:	subsection
3	:	subsubsection
	:	....

5.2 目次には 3 度のコンパイルが必要？

一度目は，目次ファイル (.toc) がありませんから，タイトル，第一節の順に文書が作成されていきます．したがって，目次ファイルには目次スペースが無い状態での頁番号が書き込まれます．二度目は，目次ファイルを読み込みますので，文書本体がその目次スペース分後ろへ順に送られます．その結果，節見出しの頁が後ろへ移動することがあるため，最悪の場合は 3 回のコンパイルが必要になるわけです．このように目次が数頁に及ぶ場合には，書籍が一般にそうであるように，目次等の前文部分と文書本体との頁番号システムを変えておくとよいでしょう．例えばこの文書では，

```
\pagenumbering{roman}
\maketitle \hrule width \textwidth
{\par\baselineskip=.8\normalbaselineskip
\begin{quotation} \tableofcontents \end{quotation} \par}
\vskip 1em \hrule width \textwidth \pagebreak
\setcounter{page}{1}\pagenumbering{arabic}
```

のようにしてあります．こうすれば，前文はローマ数字で頁番号が付され，目次の終わりで改頁すると同時に頁番号をリセットし，頁番号もアラビア数字に戻しています．

ま，しかし，それだけでもないらしく，例えば相互参照の決定・未決定の差によっても微妙に違ってきてしまう場合があるようです．コンパイル時に頁の変化があっても何もメッセージが出ないのが厄介の元です．心配な場合には毎回 .toc ファイルを保存しておいて比較して確認してください．

5.3 参考文献頁も目次に入れたい

と，この文書を作成した時に思ったので， thebibliography 環境内に次のような 1 行を加えました．

```
\addcontentsline{toc}{section}{\refname}
```

これは NTT jTEX での設定ですが， アスキー 日本語 TEX の場合もしかするとこの行の \refname の部分は明示的に『参考文献』とする必要があるかもしれません．どこであっても上と同様の行を挿入しておけば目次に出ます．出方は二番目引き数の {section} の部分に依存します．

5.4 j[-]report スタイルで目次を「第 1 章」としたい

report スタイルの章建て見出しは「第 1 章」の形式になるにもかかわらず，その目次は単に章番号だけになります．これを章建て見出しと同様にするためには次の 2 ヶ所を変更する必要があります．

まず j[-]rep10.sty 等で定義され，文字サイズによって異なる部分ですが，

```
\def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
 \refstepcounter{chapter}
 \typeout{\@chapapp{\thechapter}.}
 \addcontentsline{toc}{chapter}{\protect
 \numberline{\@chapapp{\thechapter}}#1}\else
%            ^^^^^^^^^^^^^^^^^^^^^^
 \addcontentsline{toc}{chapter}{#1}\fi
 .....
```

という部分と， j[-]report.sty で定義してある

```
\def\l@chapter#1#2{\pagebreak[3]
 \vskip 1.0em plus 1pt \@tempdima 4.5em \begingroup
%                                 ^^^^^
 \parindent \z@ \rightskip \@pnumwidth
 .....
```

の箇所です．ただ \l@chapter 定義中の幅は適当に決めましたから，もう少し適切に設定した方がエレガントでしょう．ただし アスキー 日本語 TEX では第 6.4 節 (p. 25) の変更がしてあるものとします．

5.5 book スタイルで章毎に目次を付けたい

例えば，複数の著者による文章を集めて本を作成する場合等，各章毎に目次がある方が便利でしょう．これを補助してくれるのが ‘minitoc.sty’ (ここにあります) です．文書のスタイルによって用いるコマンドが違いますが， article スタイルの場合，主目次直前に \dosecttoc を，節毎の目次を置く箇所に \secttoc を挿入するだけです．主目次が必要無い場合には \tableofcontents を \faketableofcontents に変更しておくことによって対応できます．例題とマニュアルも一緒に配布されていますので，サンプルは省略します．ただし MS-DOS 上で使用する場合には，ファイル名の部分に関し，スタイルファイル中で指示された箇所を変更しておかなければなりません．また，必ず 3 回のコンパイルが必要になります．

章建て部分について

6.1 章タイトルで直接使えないコマンドがある

いわゆる fragile なコマンドを \section{...} の中で使うとエラーになります．このことは文献 [8] にあるように，処理のタイミングの関係です．この場合には \protect コマンドをつけて壊れないようにガードする必要があります．そうした上で \string で特殊文字をエスケープしたり，空白を明示的に \space で与える必要があります．それなのに，目次を付けるような場合はうまくいかないんだな，これが．

6.2 章節タイトルをセンタリングしたい

これは文献 [2] にあったもので，章建てを定義するマクロ [‘latex.tex’] の中の

```
\def\@sect#1#2#3#4#5#6[#7]#8{\ifnum #2>\c@secnumdepth
...
```

の定義を変更します．例えばレベル 1 の章 (article では section， report では chapter) の見出しをセンタリングするには，

```
........
\ifdim \@tempskipa>\z@
  \ifnum #2=1 \begin{center} \else \fi    %<---
    \begingroup #6\relax
      \@hangfrom{\hskip #3\relax\@svsec}{\interlinepenalty \@M #8\par}
    \endgroup
  \ifnum #2=1 \end{center} \else \fi      %<---
\csname #1mark\endcsname{#7}\addcontentsline
........
```

とします． center 環境前後のグルーのため，章建て前後のスペースが少し大き目になってしまいます．同じ効果は

```
\def\section{\@startsection {section}{1}{\z@}{-3.5ex plus-1ex minus
    -.2ex}{2.3ex plus.2ex}{\reset@font\center\Large\bf}}
```

としても得られます．この前後のスペースが気になる場合は最後の方法を

```
\def\section{\@startsection {section}{1}{\z@}{-3.5ex plus-1ex minus
    -.2ex}{2.3ex plus.2ex}{\reset@font\centering\Large\bf}}
```

と修正すればいい (かもしれない) でしょう．この 2 つの差異は一体．．．未確認．

もうひとつ簡単な方法が ‘ttn2n1.sty’[29] にありました．このスタイルファイルは TEX and TUG NEWS 用のものらしいのですが，その中の定義を応用すれば，

```
\def\Section{\@ifstar{\@Section[2pt]}{\@Section[\z@]}}
%
\def\@Section[#1]#2{\ifdim #1<1pt\refstepcounter{section}\fi%
 \section*{\centering \ifdim #1<1pt%
  \addcontentsline{toc}{section}{\protect\numberline{\thesection}#2}%
   \thesection. \fi #2 \nopagebreak[4]}}
```

の定義で， \Section{...} がセンタリングした節タイトルを可能にします．注意しなければならないのは，\section ではなく \section* になっている点です．ですから \thesection も出力しなければならない上に \refstepcounter で節番号をひとつ増やす必要があります．この \refstepcounter コマンドは section 番号を増やすと同時に，それ以下の subsection 等の節番号リセットをしてくれます．上の例では \Section と \Section* とを \section コマンドと同様に使えます．前者は目次用ファイル出力も可能になっています．簡単な方法を思い付かなかったので，『長さ』で * 付きとそうでないものとを区別してみました．念のために \nopagebreak を入れて改頁の抑制をしてありますが，効果はあるでしょうか．

ちなみに，この文書の節タイトル部分は同じスタイルファイルの節見出し定義を利用しています．

6.3 章建てスタイルや前後のスペース等を変更したい

版下原稿作成では，この章タイトル前後のスペースが大き過ぎる場合があります．これは ‘art12.sty’ 等のファイルで定義されている \section マクロで調整します．例えば，英文の ‘art10.sty’ から引用すると

```
\def\section{\@startsection {section}{1}{\z@}{-3.5ex plus-1ex minus
  -.2ex}{2.3ex plus.2ex}{\reset@font\Large\bf}}
```

となっています．この \@startsection の 4, 5 番目の引数が前後のスペースを定義しています．また 4 番目の長さが負になっているのは，章の最初のパラグラフのインデントを抑制するためです．ですから，これを，

[29] ‘ttn2n1.sty’ はここにインストールされてないらしい．

```
\def\section{\@startsection {section}{1}{\z@}{1.1ex plus
 .2ex minus.1ex}{1ex plus.2ex}{\reset@font\large\bf}}
```

のようにすれば，前後のスペースが小さくなり，しかも最初のパラグラフがインデントされます．また，タイトルサイズも \large に落としてあります．もし 5 番目の引数も負にすると，章タイトルの後，改行されずに本文が続きます．(\reset@font という文字列は，'article.sty' や 'latex.tex' の古いバージョンでは用いられていないものもあります．)

さらに章建ての見出し番号表示を，例えば [1] のように変更したい場合には

```
\def\thesection{[\arabic{section}]}
```

と再定義すればいいだけです．

6.4 jreport スタイルで章建てを「第 1 章」としたい

NTT jTEX の j-report スタイルではそのようになっていますが，アスキー 日本語 TEX の場合には文献 [7] にあるような変更が必要です．つまり，自分のスタイルファイル中で

```
\def\@chapapp#1{第 #1 章}
```

とした上で (NTT jTEX では番号の位置の微調整をしていますが)，例えば 10pt の文書なら jrep10.sty の中の

```
\@chapapp{} \thechapter
```

となっている部分を含むマクロの定義箇所 (複数箇所あります) を自分のスタイルファイルにコピーした上で，この部分を

```
\@chapapp{\thechapter}
```

に変更します．これでお終いです．

6.5 'book.sty' の頁規定を緩めたい

本の場合，章の始まりはいつも右側の頁，つまり奇数頁から始まるようになっています．また目次も奇数頁から始まるようになっています．この規定を緩めるために 'book.sty' を見てみると，\tableofcontents の定義は以下のようでした．

```
\def\tableofcontents{
  \@restonecolfalse
  \if@twocolumn                % even in twocolumn
    \@restonecoltrue           % toc is in onecolumn
    \onecolumn
  \fi
  \chapter*{Contents\@mkboth{CONTENTS}{CONTENTS}}
  \@starttoc{toc}
  \if@restonecol\twocolumn\fi}  % return it into twocolumn if so
```

これを見てもわかるように，目次は \chapter* コマンドを使って表わされているので，通常の章と同様，慣習通り必ず奇数ページ (右側のページ) から始まるようになっているのです．

例えば 'bk10.sty' を見ると，\chapter の定義は次のようになっています．

```
\def\chapter{
  \cleardoublepage              % make it into odd page
  \thispagestyle{plain}         % and no header in THAT page
  \global\@topnum\z@
  \@afterindentfalse
  \secdef\@chapter\@schapter}
```

ここの \cleardoublepage を \clearpage に変更すれば頁が連続するようになります．

このような「奇数ページから始まる／始まらない」の選択をその都度できるようにするには，次のようなマクロが便利かもしれません．例えば次のマクロを 'mychap.sty' というファイルで定義します．

```
\newif\ifchaptopodd
%
\def\chapter{
  \ifchaptopodd         % If \chaptopodd=true then
    \cleardoublepage    %   \cleardoublepage
  \else                 % otherwize
    \clearpage          %   \clearpage
  \fi
  \thispagestyle{plain}
  \global\@topnum\z@
  \@afterindentfalse
  \secdef\@chapter\@schapter}
```

そうしておいて，文章中では次のように使います．

```
\documentstyle[12pt,mychap]{book}
\begin{document}

\chapter*{Prologue}
This is the prologue.

\chaptopoddfalse        % make chaptopodd false
\tableofcontents        % make the table of contents

\chaptopoddtrue         % make chaptopodd true
\chapter{AAA}
This is the chapter 1.
.....

\chaptopoddfalse
\chapter{DDD}
This is the chapter 4.
....
\end{document}
```

文書本体の作成に関連して

7.1 パラグラフ表現を変更

7.1.1 パラグラフの始まりを強調したい

ちょうど昔の西洋の本のように，パラグラフの始まりの文字だけを大きな花文字で書くようなことはできないでしょうか．

欧文の場合のこのような補助ファイルとして ‘drop.sty’ というのがあります[30]．例を示します．

```
\drop{IN} THE beginning ...
 the earth. ...
 darkness was ...
 .....
```

IN THE beginning God created the heaven and the earth. Now the earth was unformed and void, and darkness was upon the face of the deep; and the spirit of God hovered over the face of the waters.

これを漢字でもできるように改悪したものがあります．あまり面白くないですが ‘egdrop.sty’ というものです．使い方は，

```
\egdrop{\huge この \ } 遊び ....
を和文でも ....
```

この 遊びを和文でも使えるようにしたのが \egdrop{..} です．文字のサイズも指定するようにしたので，自由に場所場所で変更できます．

第 14.5.4 節 (p. 84) で示した古いドイツ花文字等を使うと面白いです．

7.1.2 パラグラフインデントあるいは左右マージンを指定したい

簡単なものなら文献 [6] の indention 環境でいいのですが，この環境中で例えば verbatim 環境等を使うと，そのマージン設定が追随しません．確かに list 環境だと簡単にできますが，行ピッチ等の微調整が面倒です．どうやら \parshape あたりを使うとよいらしく，‘indent.sty’ で定義された indentation 環境が便利に使えそうです．

コメント欄に使い方が書いてありますが
単に \begin{indentation}{30pt}{40pt} と
するだけですから，

簡単です．ふたつのオプションは左右のマージンサイズです．

7.1.3 パラグラフの形状を自分で設定したい

例えばバレンタインディに贈る手紙のパラグラフをハート型にしたいという要望があるそうです．

ハートに限らず，自分で形を作成できるようにした ‘shapepar.sty’ というスタイルファイルがあります．

```
\shapepar{\heartshape}
例えばこういうようになります．
例えばこういうようになります．
これがハート型に見えるかどうかは
難しいですが，欧文だと結構うまく
いくようです．
```

例えばこ ういうよ
うになります．例えばこういう
ようになります．これがハート
型に見えるかどうかは難しい
ですが，欧文だと結構う
まくいくようで
す．

7.1.4 パラグラフに番号を付けたい

例えば電話で草稿の打ち合わせをする場合等に，パラグラフに番号が振ってあると文章位置の指示が簡単かもしれません．これを自動的にしてくれるものに ‘numberpar.sty’ (ここにあります) があります．オプションに指定するだけですから，例は省略します．パラグラフ先頭欄外に数字が振られます．

[30] ‘drop.sty’ はインストール前にフォントサイズ指定部分を適切に設定しておく必要がある．

7.2 下線を施すことについて

7.2.1 下線で文章を強調したい

これも TEX の思想としては，あまり用いて欲しくないもののようですが，複数行にまたがらないならば通常の \underline{...} が使えます．

複数行に及ぶような場合は，欧文なら ‘ulem.sty’ が使え，日本語なら NTT jTEX 付属の ‘juline.sty’ が使えます．前者は下線だけではなく，下波線・スクラッチアウト・クロスアウトもサポートしています．また，後者をスクラッチアウトに拡張したものは ‘strike.sty’ として別に存在します．ただし JaWaTEX では前者を用いるべきですが，半角文字と漢字の部分を別々に強調した方が間違いないようです．‘ulem.sty’ の例です． When ‘**ulem.sty**’ is used, the emphasized text by **\em** is under-lined unless **\normalem** is declared at the beginning. Note that the **\verb** does not behave normally but needs **\string**. Other options are \uwave: wavy underline, \sout: ~~scratch out~~ and \xout: ~~cross out~~. ‘juline.sty’ (ここにあります) や ‘strike.sty’ (ここにあります) は \jintercharskip を \kanjiskip に変更しさえすれば アスキー 日本語 TEX でも使えるそうです．

7.2.2 文字下の波線

タイプライター時代の校正記号あるいは植字記号のようなもので，ベクトルやテンソル等のゴチック指定には下波線を使っていました．下線がイタリックだったと記憶します．最近は版下で提出することが増え，このような記号は必要なくなってきていますが，時折必要になることもあるようです．これには ‘undtilde.sty’ が使えますが，上波線があるのでそれを下ろしたもののようです． $\boldsymbol{M}$ を \undertilde{M}{} とすれば $\utilde{M}$ となります．少し tilde の大きさが足りないと思ったので \Undertilde{M}{} で $\utilde{M}$ になるようにオリジナルをちょっと変更してみました．小文字なら \undertilde{x}{} の $\utilde{x}$ でいいかな．

7.2.3 文字下の二重下線

これもタイプライター時代の校正記号でしょうが，どうしても使いたい人がいるようです．

これには ‘uuline.sty’ が使えますが， ‘ulem.sty’ の拡張機能を利用しています．古くない ‘ulem.sty’ があり，かつ拡張機能もアクティブになっている必要があります． \uuline{...} で囲むことで This sentence is doubly underlined by ‘**uuline.sty**’. のようになります．またこの ‘uuline.sty’ には，下線のような文字列の下あるいは上に任意の記号を付ける方法が紹介されています．

7.3 段組について

7.3.1 二段組と一段組を混在させたい．三段組以上にしたい

ページ単位で二段組みと一段組みを切り替えたり，あるいは二段組みページの最初の部分だけを一段組みにしたりすることは \onecolumn, \twocolumn でできますが，これはいつも改頁をしてしまいます．

ページの途中で段数を切り替えたり，三段組以上の多段組みをしたいときは ‘multicol.sty’ を使うのが一番簡単です．ただ段組環境中では，一段幅の図等のフロートが使えないなどの制約があります．

\begin{multicols}{n} の行から \end{multicols} までの間が，バランスされた n 段組になります．

```
\begin{multicols}{3}
 [\section{.....}]
```

とすれば章建ての後で段組になります．ここでは三段に分割した上で，その段の間に 0.4pt 幅の線を

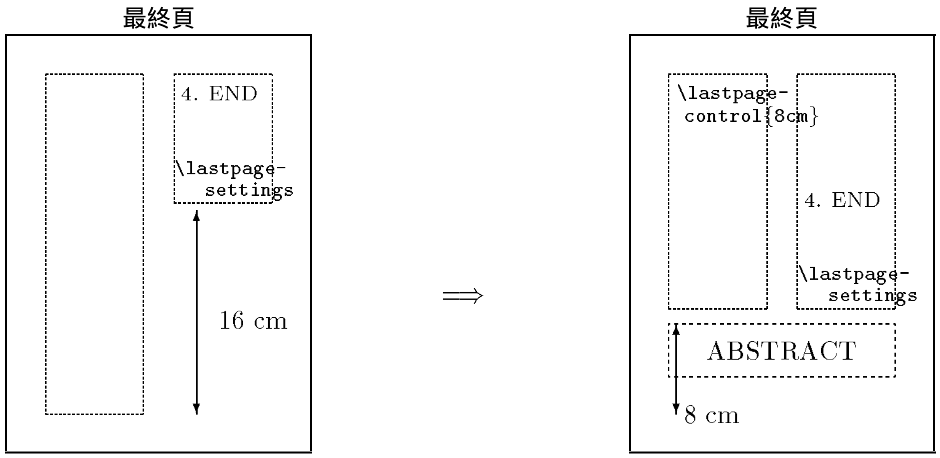


図 −**2-a** 通常の出力　　図 −**2-b** 制御したあとの出力

図 −**2** 二段組の最終頁をごまかす方法

入れてあります．この段分離線の太さは `\columnseprule` で与えますが，デフォルトは `0pt` です．ただし，もしかすると BigTEX でないとコンパイルできないことがあるかもしれません．

7.3.2 二段組の最終頁の下を揃えて，その下に一段の文章を入れたい

こういった書式が論文集に時折見られます (例えば土木学会論文集)．第 7.3.1 節 (p. 28) の ‘`multicol.sty`’ を使えば簡単ですが，このスタイルファイルにも一段内でのフロート等の問題があるようですので，ここではこのスタイルファイルを使わない少々荒っぽい方法を考えてみました．それは，最終頁の段が揃うのに必要な高さの脚注を入れ，その脚注スペースの中に `\textwidth` 幅の箱に入れた一段組文書をぶちこむ方法です．もちろん，一度出力してみてどの程度の高さが必要なのかを測定する必要がありますし，正確にというわけにはいきませんが，容認できる程度にはできそうです．不思議なことに脚注に段幅を越える箱を挿入しても警告が出ません．例を第 4.3 節 (p. 19) の例と同じく ‘`1-in-2.sty`’ (ここにあります) の中で定義してみました．スタイルファイル後半に付属している例題をコンパイルしてみてください．この例の中では，任意の頁に一段の脚注を (ひとつだけ) 作るコマンドも試してみました． この方法も土木学会用のスタイルファイルを作成しているときに思い付いたいい加減なものですが，第 3.10 節 (p. 15) の例を示したファイル ‘`clctdpp.sty`’ でも定義して使っています．最後の空白高さを自動計測して， 2 回目の実行時に調整してくれるともっといいのですが，簡単に概念を 図 −**2** に示します．

7.3.3 和文・英文等の対訳を書きたい

多くの留学生が勉学に励んでいるのが当然となった大学では，講座内の取り決め等はもとより，試験問題等も英和併記になることもあるようです．

出来具合はともかく，これに簡単に対応できそうなものに ‘`multicolpar.sty`’ がありました．これは複数の段組にしておいて，パラグラフ毎に段を変えるものです．例えば対訳では，二段組にして奇数番目のパラグラフを和文，偶数番目のを英文にすればいいわけです． おっと，‘`multicolpar.sty`’ ではなく ‘`multicol.sty`’ を読み込んだのかもしれません． `MS-DOS` では区別できないので困ったものです． ‘`multicolpar.sty`’ は無

いのかもしれません．

7.4　ルビを振りたい

これは文献 [7, 8] にあるようなマクロを組めばいいようです．

ただ NTT jTeX の場合は 'ruby.sty' が付属していますから，ファイル中のフォントサイズの定義をサイトに合わせて設定すれば，きちんとしたルビが振られます．この 'ruby.sty' もフォント名を変更すれば アスキー 日本語 TeX でも使えると思います． 使い方(つかいかた)は簡単(かんたん)で，単に `\ruby{かんたん}{簡単}` とするだけです．文献 [8] には，次に紹介するスタイルファイルと同じ様に，ルビの位置合わせを調整する方法も書いてあるようです．

あるいは 'nruby.sty' (ここにあります) というのもありますが，これはルビの文字サイズを `\tiny` に固定している代わりに アスキー 日本語 TeX, NTT jTeX 共に使えます．また傍点を付けるマクロも付いていますが，これは JaWaTeX では使えないようです．ルビとそれを振る漢字の幅を見て位置合わせをします．他に均等割り付けのマクロも付属しています．詳細はドキュメントを見てください．

7.5　イタリック補正が面倒なんですが

ローマン字体中でイタリックを使う場合，最後に `\/` で斜体によるスペース補正が必要になります．これを自動化するのが 'italic.sty'[31] です． `\it`, `\sl` に代わり `\ital{...}`, `\slant{...}` を使います．例省略．

7.6　禁則処理を変更したい

禁則文字「っ」や「ー」等の禁則処理を解除して行末の右そろえをきれいにしたいときは， NTT jTeX なら `\nokinsoku` コマンドで可能です．例えば

```
\nokinsoku`っ
```

とします．

7.7　外国語の文書作成を補助する

7.7.1　ドイツ語の文書でウムラウトが面倒

いちいちウムラウトを `\"u` 等とするのは確かに面倒です．

これには 'german.sty' があり，かなり便利になっています．例えばウムラウトは， `"u` でいいようになっています．例えば `"`"u"'` が „ü“ になり `"<"a">` が «ä» になる等です．このように他の記号についての拡張もあります．

7.7.2　フランス語の文書作成

ドイツ語の 'german.sty' と同様にフランス語のためのスタイルファイルがあります．'french.sty'[32] を使えば，フランス語入力が簡単になる (のだと思いますが，中身は私にはすべてギリシャ語！) ようです． `.fmt` ファイルを作成してフランス語用 TeX を作成するのだろうと思われます．

[31] 'italic.sty' はここにインストールされてないらしい．

[32] 'french.sty' はここにインストールされてないらしい．

7.7.3 ロシア語入力を容易にする

インストールと使い方についてはドキュメントが付属していますので省略しますが，$\mathcal{A}\mathcal{M}\mathcal{S}$-TEX のキリル文字を使うものです．一旦フォントがインストールできれば，'sscyr.sty'[33] を用いることにより，ロシア語入力が楽になるようです．

7.8 verbatim 環境の変更等

7.8.1 \verb は複数行にまたがることができない

という問題があります．例えばワークステーションのサーチパスのように長いものの場合は厄介です．そこで，適切な文字で折り返すことができるような \verb があります．それは 'path.sty' (ここにあります) というスタイルファイル[34]を使います．このファイルのオリジナルは TEX の Version 3.0 以上を対象としていますが，一部修正で，それ以下でも使えるようです．使い方は \verb と同じですから割愛します．もちろん \verb ではなく \tt で書いた上で，/ を \slash で書くように心掛ければ [8] 何も問題はありませんが，この \path は / 以外の文字も区切りにしてくれるようです．

7.8.2 verbatim 環境中にコマンドを入れたい

verbatim 環境中で \ は文字として出力されるため，文字の大きさを途中で変更したりすることはできません．また JaWaTEX を使っている場合には verbatim 環境中に漢字を入れることができません．

これに対処するためのスタイルファイルとして 'alltt.sty' というのがあります[35]．ファイル名が示すように文字を Typewriter Type にしたまま \ や {} 等を TEX の特殊文字として扱う環境を作るものです．この環境を使えば，verbatim に似た出力のまま文字の大きさを途中で変更したりできるようになります．

```
\begin{alltt}
 This is an example of ....
 \large 'alltt.sty'. ...
 ...
 \scriptsize This is the end...
\end{alltt}
```

This is an example of the environment
'alltt.sty'. ここからその環境です．
ここにもあるようですが，
使い方はスタイルファイルにも書いて
ありますから，割愛します．
This is the end of example.

同様のものに 'astyped.sty'[36] というのがありますが，使ったことが無いので説明や例は省略します．

7.9 草稿と清書とで内容を変える．あるいは複数行の無視

文書作成中に思い付いたことを忘れないように本文に書き込んだが，差し当たってその部分を処理したくない場合があります．あるいは，最終版と草稿とで異なる記述をしたい部分もあるかもしれません．このような文章選択指定を簡単にするためのスタイルファイルとして 'version.sty' (ここにあります) というのがあります．しかし一番の使い道は先に述べたように，何行にもまたがる部分をとりあえず無視する場合等に，例えば \begin{comment}, \end{comment} で囲めばよいということです．例えば草稿版を書いておく draft 環境を \includeversion{draft} で，清書版の方を \excludeversion{final} で宣言しておく (後者のデ

[33] 'sscyr.sty' はここにインストールされてないらしい．

[34] akiu:ftp/pub/tex/latex-styles/ にあるものは，Version 3.0 以下の TEX でも使えるように修正されている．TEX のバージョンが新しい場合には該当する部分を元に戻して使うこと．

[35] 'example.sty' というのもあったが，近年の 'example.sty' はこの 'alltt.sty' の機能以上のことをするものに変更されているようなので，ここでは推奨しない．

[36] 'astyped.sty' はここにインストールされてないらしい．

フォールトが comment になっている) と，通常は draft 環境に含まれた部分だけが処理・出力され， final 環境の部分は無視されます．清書の段階では \includeversion と \excludeversion とを入れ替えれば，逆の状態になります．

'version.sty' はこの文書パッケージに含まれていますが，それは，紹介する各スタイルファイルのインストール状況に応じて例題を示す場合とそうでない場合の文章を用意する必要があったからです．

複数行の無視には 'comment.sty'[37] というのもあります． 'version.sty' とほとんど同じで，そのコンパイルで含める部分とそうでない部分の環境名を \includecomment{...} と \excludecomment{...} とで定義します．

7.10 脚注の処理や変更について

7.10.1 脚注や図表を文書のうしろにまとめたい

論文のマニュスクリプト等にあるように，図表をうしろにまとめたい場合があります．また，脚注が好ましくないことがあり，説明を文章の最後にまとめる書式のものもあります．このような要求に対応するための補助スタイルファイルがあります．ひとつは 'endnote.sty'[38] です． もうひとつは 'endnotes.sty'[39] です．図表については 'endfloat.sty'[40] です．使ってないので，説明は省略します．

7.10.2 短い脚注を横並びにしたい

ほとんどの脚注が文書幅に比べて比較的短い時，通常の文書幅一杯の脚注は間が抜けて見えます．脚注を順に横に並べることができれば，多少見難いですがスペースは節約できそうです．これをするのが 'fnpara.sty' (ここにあります) です．オプション指定すれば結果は明白ですから，説明は省略します．

7.10.3 長い脚注にインデントをつけたい

脚注が長くなると脚注参照記号の下に文章がくるので多少見難くなります．好みの問題かもしれませんが，インデントした方が読み易くなります．これは JaWaTEX の作者渡辺氏[41]の定義で

```
\long\def\@makefntext#1{\parindent 1em\noindent
  \hbox to 2em{\hss$^{\@thefnmark}$~}%
  \@tempdima\columnwidth\advance\@tempdima-2em\parbox[t]{\@tempdima}{#1}}
```

と変更すればそれができ，この文書でも使ってあります．複数行に及ぶ脚注をご覧ください．また，稲村氏による同様の定義を 'hanging.tex'[42] に保存してあります．

7.10.4 tabular 環境等の脚注が出力されない

マニュアル等に \footnotemark, \footnotetext{...} および \addtocounter{footnote}{??} を使うように，と書いてあるにも拘わらず，よく質問されるもののひとつです．そういった環境中の脚注も通常のも

[37] 'comment.sty' はここにインストールされてないらしい．

[38] 'endnote.sty' はここにインストールされてないらしい．

[39] 'endnotes.sty' はここにインストールされてないらしい．

[40] 'endfloat.sty' はここにインストールされてないらしい．

[41] m.watanabe@tainsbbms, watanabe@akiu.gw.tohoku.ac.jp

[42] 'hanging.tex' はここにインストールされてないらしい．

のと同様に出力されるようにするものに ‘ftn.sty’ (ここにあります) があります．脚注を \footnote の代わりに \ftn{...} で定義するだけですが，使用していないので説明・例を省略します．

7.10.5　二段組での脚注をすべて右段に集めたい

というような希望があるようです．そういえば ASME の論文集ではそうなっていたかもしれません．これには Mainz ファイルの ‘ftnright.sty’[43] があります． 第 7.3.1 節 (p. 28) の段組環境 ‘multicol.sty’ とは共存できないようです．マニュアル付きですので，例は省略します．

7.11　ハイフンの例外を指定したい

TeX 自体が持っている規則は必ずしも正しいハイフンを付けるとは限らないようです．そのような場合はその都度指定すべきでしょう．それには

```
\hyphenation{man-u-script man-u-scripts ....
             ....}
```

のようにします．公になっているコレクションもたくさんあるようですから，そういった例外集[44]を探した方が確実かもしれません．

7.12　部分的に改行ピッチを変更したい

この文書でも用いていますが，例えばプログラムリストや英文を和文中に含めたい場合に，改行ピッチをその部分だけ小さくしたいことがあります．これには \baselineskip を変更すればいいのですが，厄介なことに \baselineskip は改行に出会って初めて有効になりますので，途中から変更する場合でも前後に改行が入ってしまいます．例えばここでは english 環境と呼んで定義してみますと，

```
\def\english{\par\baselineskip=.7\normalbaselineskip
  \vskip .4286\baselineskip\noindent\ignorespaces}    % .4286=(1-.7)/.7
\def\endenglish{\par}
```

という宣言のもとで

```
\begin{english}
Programs may be put here...
....
\end{english}
\noindent where ....
```

のように使います．出力例はこの文書の随所にあります．注意することは \begin{english} は改行しますがインデントはせず， \end{english} は改行もインデントもしますから，上の例のように文章を続けるばあいには \noindent が必要となります．

通常，文全体の改行幅は \baselinestretch の変更で簡単に行えますが，この方法は文書中途でも使えるでしょうか．(ここで改行幅を二倍にしてみます)

TeX and TUG NEWS (Vol.2, No.3 June 1993 ‘ttn2n3.tex’) に掲載されていた記事によれば，この変更

の直後に文字サイズを変更すれば設定が生きるようです．つまり，

[43] ‘ftnright.sty’ はここにインストールされてないらしい．

[44] akiu:ftp/pub/tex/latex-styles/bear_collections にひとつあります．

```
\renewcommand{\baselinestretch}{2}\tiny\normalsize
```

の一行で，この次の行からダブルスペースの普通サイズ文字になります．(元の改行幅に戻します)

上のパラグラフ内改行幅をこの方法で変更してみましたが，この変更を行う位置前後の改行幅については，この例でも明らかなように，目的に応じてある程度細かな設定が必要かもしれません．

7.13 プログラムリストを書きたい

いちばん簡単なのは verbatim 環境を使うことで，行間隔も小さくしたいですから，これは上の english 環境のようにすればいいでしょう．あるいは，

```
\newenvironment{program}%
  {\begin{quote}\addtolength{\baselineskip}{-0.8ex}}%
  {\end{quote}}
```

ということも同じことです．ここでは quote 分のインデントを含めてあります．

ただ，別ファイルになっているプログラムリストをわざわざその文書ファイルに挿入するのは面倒です．

この場合には ‘verbatimfiles.sty’ が使えます．\verbatimfile{filename} でファイルの内容がそのまま verbatim 環境で出力されます．また \verbatimlisting{filename} で行番号付きになります．

左が前者で，右が番号付きの後者のコマンドを使ったものです．

```
can write only FORTRAN programs
      program style
      dimension a(10,10),....
      ......
      do 10 i=1,n
      ......
   10 continue
      ......
could be any text files.
```

```
1  can write only FORTRAN programs
2        program style
3        dimension a(10,10),....
4        ......
5        do 10 i=1,n
6        ......
7     10 continue
8        ......
9  could be any text files.
```

また C のプログラムであれば ‘cprog.sty’[45] というのもあります．使ったことが無いので説明は省略します．

FORTRAN や Prolog 等のプログラムリストを LaTeX に変換し，キーワードや予約語等をフォントを変えて表示するプログラムがあるようです．‘lgrind’ というのがそれで， plain TeX への変換は ‘tgrind’ というもののようです． C や FORTRAN をはじめ， CSH, Mlisp 等 25 種類ほどのプログラムや記述ファイルの変換ができるようです．ただ，コメント欄の空白を変換後に ~ で置き換えないと左に詰まってしまいます．

7.14 プログラム等のアルゴリズム表示

定理環境のようなものですが，プログラムのフローを If 等を使って分かりやすく表示したい場合があります．これには ‘algorithms.sty’[46] が使えます．ただし， ‘algorithm.sty’ が存在するため， MS-DOS では区別がつかず問題があります．使い方はサンプルを参照してください．数学の所で紹介した ‘proof.sty’ とは用途が違います．

[45] ‘cprog.sty’ はここにインストールされてないらしい．

[46] ‘algorithms.sty’ はここにインストールされてないらしい．

7.15 太い罫線で見出しにアクセントを

TEX and TUG NEWS (Vol.2, No.4 Oct. 1993 'ttn2n4.tex') の求人欄にあったのですが，こんな使い方はプロには許されても，通常はやめておいた方がいいのかもしれません．というわけで，改悪した結果だけを示しておきます．あまり多用しないでください．

——— 本日の会議案内 ———

丸○大学特別選抜入学判定会議 — 特別選抜検討委員会

本日午後△時より，□館第七会議室において，表記委員会の重要会議を開催します．なお，今回の会議においては．．．．．

1. ▽◎に関する取り扱い注意資料を持参ください
2. またも退屈な会議になりそうです

というのが次の announce 環境で出力されます．

```
\def\REYrule{\hbox to 2cm{\leaders\hrule height 3pt\hfill}}
\newbox\REYbox
\def\announce#1{%
  \setbox\REYbox=\hbox{\REYrule\quad{\LARGE\bf #1}\quad\raise3pt\REYrule}%
  \gdef\REYbigrule{\hbox to \wd\REYbox{\leaders\hrule height 3pt\hfill}}%
  \vspace{1em}%
  \centerline{\raise3pt\REYrule\quad{\LARGE\bf #1}\quad\raise3pt\REYrule}}
\def\endannounce{\par\centerline{\REYbigrule}}
```

上の例は \begin{announce}{本日の会議案内} で始めました．なお，この文書の目次の後にある注意書きも同様の環境で作成してみましたが，そちらは罫線の長さも指定できるように変更してあります．

7.16 他の環境

アルゴリズムを書くのに便利な 'algorithm.sty'[47]， プログラム表示の補助をする 'program.sty'[48] 等があります．使い方・例については省略します．

数式に関連して

8.1 式番号の付け方について

8.1.1 数式番号に副番号をつけたい

LATEX では式番号が自動的に振られますが，カウンタ equation は数式環境の \nonumber 無しの行毎に増加する一方です．特に eqnarray 環境等で副番号が例えば (3-a) 等のように付けば便利です．ひとつの方法は 'subeqn.sty'[49] を用いることです．これだと式と式の間にテキストがあるような場合でも副番号を継続するこ

[47] 'algorithm.sty' はここにインストールされてないらしい．

[48] 'program.sty' はここにインストールされてないらしい．

[49] 'subeqn.sty' はここにインストールされてないらしい．

とができます．使っていないので説明は省略します． また ‘subeqnarray.sty’[50] というものもありますが，使っていないので説明は省略します．使っていない理由は，\label がうまく作動しないような印象があるからです．

これに対し，簡便なスタイルファイルを自作してみました．

それは ‘manyeqns.sty’ というファイルですが，単に副番号のカウンタを定義し，新しい manyeqns 環境毎にリセットしているだけです．手動でなら副番号の継続等も可能です．また，\ref による引用も簡単で，各式だけでなくそれを総合したグループ式番号も引用できるようにしてあります．例えば次のようになります．

```
\begin{manyeqns}
\label{eq:thiseq}
 F & = & \int_\Gamma \sin z \, dz \label{eq:thiseq1} \\
 G & = & \sum_{n=0}^{\infty} a_n \, t^n \label{eq:thiseq2}
\end{manyeqns}

ここに ..... 式 (\ref{eq:thiseq2}) は ..... となるから
式 (\ref{eq:thiseq1}) を ..... つまり式 (\ref{eq:thiseq}) で
示された .....

     \addtocounter{equation}{-1}      % <----- sets previous eq. number
\begin{manyeqns}
     \addtocounter{whichequation}{2}  % <----- skips "a" and "b"
 H & \equiv & \int_0^\ell g(z)\,\sin\,z\,dz
\end{manyeqns}

とすれば，間に文章があっても続く式番号になる．
```

$$F = \int_\Gamma \sin z \, dz \qquad (3\text{-a})$$

$$G = \sum_{n=0}^{\infty} a_n \, t^n \qquad (3\text{-b})$$

ここに ...式 (3-b) は ...となるから式 (3-a) を ...つまり式 (3) で示された ...

$$H \equiv \int_0^\ell g(z)\,\sin\,z\,dz \qquad (3\text{-c})$$

とすれば，間に文章があっても続く式番号になる．

\addtocounter{equation}{-1} はカウンタを増減するものです．ここでは式番号をひとつ前のと揃え，副番号を前の続きにするために使ってあります．

やはり ‘subeqnarray.sty’ 等には \label の件で不満があるようです．

ここに紹介するのは plain TeX の \case のように使う複数行の副番号付きの数式環境で，スタイルファイルは ‘cases.sty’ です．元ファイルに付属していたサンプルのひとつを下に付けました．

```
\begin{subnumcases}{f(x)=\label{f(x)}}
1/3 & if $0 < x < 1$;\\
2/3 & if $3 < x < 4$;\label{.6666}\\
0   & elsewhere.
\end{subnumcases}
```

$$f(x) = \begin{cases} 1/3 & \text{if } 0 < x < 1; \qquad (4a) \\ 2/3 & \text{if } 3 < x < 4; \qquad (4b) \\ 0 & \text{elsewhere.} \qquad (4c) \end{cases}$$

[50] ‘subeqnarray.sty’ はここにインストールされてないらしい．

8.1.2 数式の番号に節の番号を添えたい

節や章毎に数式カウンタをリセットし，数式番号を (節. 番号) のようにするには，`\theequation` の定義を変更すればよいだけです [‘`latex.tex`’] が，次のようなことも可能です．

```
\@addtoreset{equation}{section}
\def\theequation{\thesection 節の \arabic{equation} 式}
```

$$F = \int_\Gamma \sin z \, dz \qquad \text{(8 節の 5 式)}$$

この `\@addtoreset{cA}{cB}` というのは，カウンタ `cB` が変化する時に `cA` を零にリセットするというものです．

8.1.3 複数行の式にひとつだけ式番号を与えたい

もちろん `eqnarray` 環境に `\nonumber` を使えばいいことですが，偶数行の場合の位置には不満があります．そこで `equation` 環境に `array` 環境で式を並べてみました．ただ，デフォールトでは `\textstyle` になりますので，積分や総和を入れる場合には明示的に `\displaystyle` とする必要があります．

```
\begin{equation}
\begin{array}{l}
{\displaystyle I=\sum_{i=1}^N a_i\,b_i} \\
               \noalign{\vskip 1ex}
{\displaystyle f=\int_0^\ell g(x)\,\sin x\, dx}
\end{array}
\end{equation}
```

$$\begin{array}{l} {\displaystyle I=\sum_{i=1}^N a_i\,b_i} \\ {\displaystyle f=\int_0^\ell g(x)\,\sin x\, dx} \end{array} \tag{6}$$

この例のように `\int` 等を使う場合には `\noalign` を使って行間隔を制御する必要があるかもしれません；第 8.4.1 節 (p. 38) 参照．

8.2 左寄せの式にリーダーを付けたい

式を左寄せにした (‘`fleqn.sty`’ を用いた) 場合，式番号と式の間が広くなると見難いものです．そのために，論文集等ではリーダーを付けることが多いようです．

ここで示すマクロは磯崎氏 [8] と渡辺氏の定義を用いたものです．スタイルファイル名は ‘`doteqn-f.sty`’ がそれに当たります． `eqnarray` 環境にリーダーを付けます．

```
\begin{eqnarray}
 F & = & \int_\Gamma \sin z \, dz  \\
 G & = & \sum_{n=0}^{\infty} a_n \, t^n
\end{eqnarray}
```

$$F = \int_\Gamma \sin z \, dz \cdots\cdots\cdots (7)$$

$$G = \sum_{n=0}^{\infty} a_n \, t^n \cdots\cdots\cdots (8)$$

通常の `equation` 環境にもリーダーを付すことについては，上で説明した ‘`manyeqns.sty`’ のパッケージ中にある渡辺さんのマクロを含めた ‘`dot-eqns.sty`’[51] を用いればいいでしょう．特に設定は必要ありませんので例は割愛します．

[51] ‘`dot-eqns.sty`’ はここにインストールされてないらしい．

8.3 eqnarray で & 相当部の空白が広すぎる

これについては，eqnarray 環境の定義中の次の行を変更すればいいようです [7]．

```
....
$\displaystyle\tabskip\z@{##}$&\global\@eqcnt\@ne
 \hskip 2\arraycolsep \hfil${##}$\hfil
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
&\global\@eqcnt\tw@ \hskip 2\arraycolsep $\displaystyle\tabskip\z@{##}$\hfil
%                    ^^^^^^^^^^^^^^^^^^^^^^
 \tabskip\@centering&\llap{##}\tabskip\z@\cr}
```

となっている部分を，例えば

```
$\displaystyle\tabskip\z@{##}$&\global\@eqcnt\@ne
 \hfil$\;{##}\;$\hfil
&\global\@eqcnt\tw@ $\displaystyle\tabskip\z@{##}$\hfil
 \tabskip\@centering&\llap{##}\tabskip\z@\cr}
```

のように変更します．文献 [7] ではさらに，この 2 つの & で挟まれた部分も \displaystyle に変更してありますが，通常，その必要は無いでしょう．

$$\begin{array}{rcl} F & = & \displaystyle\int_\Gamma \sin z\,dz \\ G & = & \displaystyle\sum_{n=0}^\infty a_n\,t^n \end{array} \quad\longrightarrow\quad \begin{array}{r@{}c@{}l} F & = & \displaystyle\int_\Gamma \sin z\,dz \\ G & = & \displaystyle\sum_{n=0}^\infty a_n\,t^n \end{array}$$

8.4 行列に関連して

8.4.1 行列の行間の詰まりすぎを防止する，あるいは式の間に文章を挟む

行列の中に displaystyle の分数などが入ると，行と行が重なってしまうことがあります．これを離すには \noalign{\vskip ...} を用います [4]．また eqnarray 環境で式を並べている時に，その環境の中に直接数行の文章を挿入したい場合等の行間隔制御も同様にできます．例えば

```
\left\{\begin{array}{c}
 f_1\\ f_2\\ f_3
\end{array}\right\}=
\left(\begin{array}{ccc}
 A & 0 & B \\
       \noalign{\vskip 1.2ex}  % <-----
 0 & \displaystyle{\frac{EA}{\ell}} &
 -\displaystyle{\frac{EA}{\ell}}\\
       \noalign{\vskip 1.2ex}  % <-----
 B & -\displaystyle{\frac{EA}{\ell}} &
 \displaystyle{\frac{EA}{\ell}}
\end{array}\right)\left\{
\begin{array}{c}
 u_1\\ u_2\\ u_3
\end{array}\right\}
```

$$\left\{\begin{array}{c} f_1 \\ \hdashline f_2 \\ f_3 \end{array}\right\} = \left(\begin{array}{c:cc} A & 0 & B \\ \hdashline 0 & \dfrac{EA}{\ell} & -\dfrac{EA}{\ell} \\ B & -\dfrac{EA}{\ell} & \dfrac{EA}{\ell} \end{array}\right) \left\{\begin{array}{c} u_1 \\ \hdashline u_2 \\ u_3 \end{array}\right\}$$

となります．ここの破線については第 10.7 節 (p. 57) を参照のこと．

8.4.2 二次形式の行列表示等の高さ揃えはどうする

二次形式の行列表示のように，行ベクトルを行列の上部に揃えたい場合には，いくつかの方法があるようです．主に plain TEX のコマンドの修正版で，例えば Daniel H. Luecking 氏のマクロであれば

```
\def\tpmatrix#1{
 \setbox\z@=\vtop{\normalbaselines\m@th
  \ialign{\hfil$##$\hfil&&\quad\hfil$##$\hfil\crcr
   \mathstrut\crcr
   \noalign{\kern-\baselineskip}
   #1\crcr
   \mathstrut\crcr
   \noalign{\kern-\baselineskip}
  }
 }
 \dimen\z@=\dp\z@
 \setbox\z@=\vbox to \ht\z@{
  \hbox{$\displaystyle \left(\,\vcenter{\unvbox\z@}\,\right)$}
  \vss
 }
 \dp\z@=\dimen\z@
 \box\z@
}
```

というマクロを定義しておけば，

```
\pmatrix{x & y & z}
\tpmatrix{a_1 & a_2 & a_3 \cr
     a_4 & a_5 & a_6 \cr
     a_7 & a_8 & a_9 \cr
}
\tpmatrix{b \cr c \cr d}
```

$$\begin{pmatrix} x & y & z \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix} \begin{pmatrix} b \\ c \\ d \end{pmatrix}$$

もうひとつの方法は Donald Arseneau 氏によるもので，

```
\newbox\matbox
\def\topmatrix#1{\setbox\matbox=\vtop{\normalbaselines\m@th % set the matrix in
  \ialign{\hfil$##$\hfil&&\quad\hfil$##$\hfil\crcr    % a \vtop so the
  \mathstrut\crcr\noalign{\kern-\baselineskip}      % first baseline
  #1\crcr\mathstrut\crcr\noalign{\kern-\baselineskip}}} % lines up.
% get twice difference between baseline and centerline of inner matrix:
  \dimen255=\dp\matbox \advance\dimen255 by -\ht\matbox
% Center matrix and surround with parentheses:
  \setbox\matbox=\hbox{$\left( \,\vcenter{\box\matbox}\,\right)$}
% Correct for difference between baseline and centerline of parentheses:
  \advance\dimen255 by -\dp\matbox \advance\dimen255 by \ht\matbox
% Lower centered matrix back to its proper baseline:
  \lower0.5\dimen255\box\matbox
}
```

に対し，同様に

```
\pmatrix{x & y & z}
\topmatrix{a_1 & a_2 & a_3 \cr
          a_4  & a_5 & a_6 \cr
          a_7  & a_8 & a_9 \cr
}
\topmatrix{b \cr c \cr d}
```

$$\begin{pmatrix} x & y & z \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix} \begin{pmatrix} b \\ c \\ d \end{pmatrix}$$

最後は Bruce Ikenaga 氏による一番簡単な方法で，

```
\matrix{ A = \cr
  \vphantom{ 0 } \cr
  \vphantom{ 0 } \cr}
\matrix{ \pmatrix{a & b & c\cr} \cr
  \vphantom{ 0 } \cr
  \vphantom{ 0 } \cr}
\pmatrix{ 1 & 0 & 0 \cr
  0 & 1 & 0 \cr
  0 & 0 & 1 \cr}
\pmatrix{b \cr c \cr d}
```

$$A = \begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b \\ c \\ d \end{pmatrix}$$

というものです．これを LaTeX で真似すると

```
\begin{array}{ccc}
\lfloor\quad a & b & c \quad\rfloor \\
\vphantom{0} \\
\vphantom{0}
\end{array}
\left(\begin{array}{ccc}
  a_1 & a_2 & a_3 \\
  a_4 & a_5 & a_6 \\
  a_7 & a_8 & a_9
\end{array}\right)
\left\{\begin{array}{c}
x \\ y \\ z
\end{array}\right\}
```

$$\begin{array}{ccc} \lfloor\quad a & b & c \quad\rfloor \\ \vphantom{0} \\ \vphantom{0} \end{array} \left(\begin{array}{ccc} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{array}\right) \left\{\begin{array}{c} x \\ y \\ z \end{array}\right\}$$

となります[52]．

8.4.3 行列表示等を簡単化

行列や場合分けの表示，あるいは行列の縦方向の位置合わせを簡単にできないでしょうか．

それには ‘delarray.sty’ がよさそうです．使い方が含まれていますので実行してみてください．ただし常に ‘array.sty’ が必要です．ここにはマニュアルからそのままの例示をしておきます．

```
\begin{array}({cc})
a_{11} & a_{12} \\
a_{21} & a_{22} \end{array}
```

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

[52] 関連した元記事は ‘tpmatrix.tex’ の中．

```
\newcolumntype{L}{>{$}l<{$}}
f(x)=\begin{array}\{{lL}.
 0 & if $x=0$ \\
 \sin(x)/x & otherwise \end{array}
```

$$f(x) = \left\{ \begin{array}{ll} 0 & \text{if } x=0 \\ \sin(x)/x & \text{otherwise} \end{array} \right.$$

```
 \begin{array}[t]\{{c}\}
   3 \\ 4 \\ 5 \end{array}
 \begin{array}[c]\{{c}\}
   2 \\ 3 \\ 4 \end{array}
 \begin{array}[b]\{{c}\}
   1 \\ 2 \\ 3 \end{array}
```

$$\left\{\begin{array}[t]{c} 3 \\ 4 \\ 5 \end{array}\right\} \left\{\begin{array}[c]{c} 2 \\ 3 \\ 4 \end{array}\right\} \left\{\begin{array}[b]{c} 1 \\ 2 \\ 3 \end{array}\right\}$$

8.5　式を上下ブレースで分割する

ちょっと説明するのは難しいので，例示します．

スタイルファイルとしては 'ovudbrac.sty' としていますが，元々はそういう名称ではありません．作者は前出の Donald Arseneau 氏です．

```
                 x               y
           ,-----^-----,   ,-----^-----,
   a + b + c + d + e + f + g + h + i + j + k + l + m = \pi r^2
                   `-----v-----'
                         z
```

といった風のことをするためのマクロです．

```
\overunderbraces{&\br{2}{x}& &\br{2}{y}}%
   {a + b +&c + d +&e + f&+&g + h&+ i + j&+ k + l + m}%
                {& &\br{3}{z}} = \pi r^2
```

$$a+b+\overbrace{c+d+\underbrace{e+f}^{x}+\overbrace{g+h}}_{z}^{y}+i+j+k+l+m=\pi r^2 \tag{9}$$

8.6　空白調整と括弧の高さ等

数式モードでは空白を手動で決める必要があります．格好よくするのは結構面倒なものです．目安のために空白の広さを比較しておきます．

```
g=\int_a^b\int_c^e f(x,y)\,dx\,dy
```

$$g \quad = \quad \int_a^b\int_c^e f(x,y)\,dx\,dy$$

```
g=\int_a^b\!\!\!\int_c^e f(x,y)\,dx\,dy
```

$$g \quad = \quad \int_a^b\!\!\!\int_c^e f(x,y)\,dx\,dy$$

```
f(x)=a  x + b
f(x)=a\,x + b
f(x)=a\:x + b
f(x)=a\;x + b
```

$$\begin{aligned} f(x) &= ax+b \\ f(x) &= a\,x+b \\ f(x) &= a\:x+b \\ f(x) &= a\;x+b \end{aligned}$$

また括弧の高さを手動で行なう場合のために

```
\left\{\sqrt{1+\frac{\pi}{\omega}}
\Biggl[\biggl(\Bigr|\bigr](a+b)\right\}
```

$$\left\{\sqrt{1+\frac{\pi}{\omega}}\Biggl[\biggl(\Bigr|\bigr](a+b)\right\}$$

さらに記号の積み上げについても必要ですが，文献 [4] 等からその例だけを示しておきます．

```
g\stackrel{\rm def}{=}\sin(x)...
\sum_{{\scriptstyle j=1}
 \atop{\scriptstyle i\neq j}}^n
\mathop{\bf Res}_{\Im z>0}\,f(z)
\mathop{\sum_{i=1}^n \sum_{j=1}^m}%
      _{i+j=\mbox{\scriptsize even}}
```

$$g\stackrel{\rm def}{=}\sin(x)..., \quad \sum_{{\scriptstyle j=1}\atop{\scriptstyle i\neq j}}^n, \quad \mathop{\bf Res}_{\Im z>0}\,f(z), \quad \mathop{\sum_{i=1}^n \sum_{j=1}^m}_{i+j=\mbox{even}} \tag{10}$$

8.7 ちょっとだけ plain TEX (かな?) の数式モード

8.7.1 ルートの高さ揃え

```
a &=& \sqrt{g}+\sqrt{h}
b &=& \sqrt{\mathstrut g}+\sqrt{\mathstrut h}
```

$$\begin{aligned} a &= \sqrt{g}+\sqrt{h} \\ b &= \sqrt{\mathstrut g}+\sqrt{\mathstrut h} \end{aligned}$$

8.7.2 上下に重ねる

```
{a \atop b} + c
a \atopwithdelims<> b
a \atopwithdelims\{. b
```

$${a \atop b} + c, \qquad a \atopwithdelims<> b, \qquad a \atopwithdelims\{. b$$

8.7.3 行列

```
A=\pmatrix{ a & b \cr
           c & d \cr }
A= \left[ \matrix{ a & b \cr
          c & d \cr } \right]
```

$$A=\begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad A=\left[\begin{matrix} a & b \\ c & d \end{matrix}\right]$$

8.7.4 場合分け

```
|x|=\cases{ x & (if $x$ $\ge$ 0) \cr
          -x & (otherwise) \cr }
```

$$|x|=\begin{cases} x & (x \ge 0 \text{ のとき}) \\ -x & (\text{それ以外のとき}) \end{cases}$$

8.8 文章中で実数計算をしたい

TEX の計算は整数でやっているようですから実数計算は難しいようです．

そこで，これには磯崎氏の `eclarith.sty' が便利です．例えば \Sqroot{2.0}\Sqroottwo とすれば，\Sqroottwo が 1.414215 となり，その結果の 1.5 倍なら \Mul{1.5}\Sqroottwo\Ans として \Ans を印字すれば 2.12132 となります．これ以外に多種計算や描画等に便利なマクロが含まれています．

8.9 化学式を簡単に書きたい

これはなかなか難しいようです．地道にマクロを書くしかないかもしれません． $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX ではかなり簡単にかけるようになっているようですが未経験です．また構造式を書くのはもっと難しいようで，ChemTeX があり，ある程度のことはできるようです．`chemtex.sty'[53] を使います．使い方マニュアルも付属しています[54]から，ここでは説明を省略します．また反応式なら `dchem.sty' というのがあり，数式の equation や eqnarray に相当する環境が定義されています．

```
\begin{reaction}
 H + O_2
  \eqbm^{\cit k\sub{a}}_{\cit k\sub{-a}} HO_2
\end{reaction}
```

例えば reaction 環境なら

$$H + O_2 \underset{k_{-\mathrm{a}}}{\overset{k_{\mathrm{a}}}{\rightleftharpoons}} HO_2 \qquad (11)$$

となります．他はコメント欄を参照してください．

さらに plain TeX 用の補助ファイルもあります．ファイル名は `chemstruct.tex'[55] ですが，もちろん使ったことは無いので説明は省略します．

8.10 可変長の右矢印はありませんか

矢印を長く伸ばし，その上下に文字列を置きたい場合があります．これは前節の `dchem.sty' の中で定義されているマクロ \Yields が数式モードで使えます．例えば $A \xrightarrow[\text{This is a long text.}]{f(x)=\sin(x)} B$ は

```
$A \Yields_{\mbox{\scriptsize This is a long text.}}^{f(x)=\sin(x)}B$
```

の出力です．長さを \Yields[3cm] 等と自分で指定することもできます．長さ指定なら両方向の矢印 \Eqbm も使えます．詳細はコメント欄を参照のこと．

8.11 二重鉤括弧

これも TeX and TUG NEWS (Vol.1, No.2 May 1992 `ttn1n2.tex') にあったものですが，Michael Barr 氏 (McGill Univ., Montreal, barr@math.mcgill.ca) によるマクロで，

```
\newdimen\argwidth
\def\[[#1\]]{%
    \setbox0=\hbox{$#1$}\argwidth=\wd0
    \setbox0=\hbox{$\left[\box0\right]$}\advance\argwidth by -\wd0
    \left[\kern.3\argwidth\box0\kern.3\argwidth\right]}
```

[53] `chemtex.sty' はここにインストールされてないらしい．

[54] UNIX 上のコンパイルでは，マニュアルファイル名の大文字・小文字の修正が必要．

[55] `chemstruct.tex' はここにインストールされてないらしい．

の定義により，$\[[\sum_{i\in I}A_i=A\]]$ が $\llbracket\sum_{i\in I} A_i = A\rrbracket$ と表示されます．数式環境なら鈎括弧の中の部分で \displaystyle を指定しておけば，

$$\left\llbracket\sum_{i\in I} A_i = A\right\rrbracket \tag{12}$$

のようになります．

8.12 $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX の記号を LATEX で使いたい

数学用のフォントを豊富に揃えた $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX の記号を使いたい場合には 'amssymbols.sty'[56] を使いますが，使ったことがないので使い方は省略します．

8.13 積分記号の変更や改訂

8.13.1 より大きな，あるいは直立した積分記号

積分記号は被積分関数の高さに合わせて大きくなりますが，どこまでもという訳にはいかないようです．あるいは通常の少し傾いた記号を好まない方もおられるようです．

これについては 'bigint.sty' を使えば対応できます．ほとんど通常の \int と同じ使い方ですが，

```
\bigint[overlap symbol] [_lower][^upper] { integrand }
```

のように使います．鈎括弧部分はオプションです．被積分関数を括弧で囲む必要があるのと，オプションで積分記号に重ねる記号を設定できる点が通常の積分記号と異なります．その差は

```
\bigint_a^b {\frac{\displaystyle\frac{c}{x^2}}{\sqrt{\displaystyle
                                        d + \frac{c}{x}}} \, dx} = e + ...
\bigint[\bigcirc]_\Gamma {\frac{\displaystyle\frac{c}{z^2}}{\sqrt{
                              \displaystyle d + \frac{c}{z}}} \, dz} = e + ...
```

$$\int_a^b \frac{\dfrac{c}{x^2}}{\sqrt{d+\dfrac{c}{x}}}\,dx = e+\cdots, \qquad \int_a^b \frac{\dfrac{c}{x^2}}{\sqrt{d+\dfrac{c}{x}}}\,dx = e+\cdots,$$

$$\oint_\Gamma \frac{\dfrac{c}{z^2}}{\sqrt{d+\dfrac{c}{z}}}\,dz = e+\cdots, \qquad \oint_\Gamma \frac{\dfrac{c}{z^2}}{\sqrt{d+\dfrac{c}{z}}}\,dz = e+\cdots$$

のように対比すれば明らかです．

8.13.2 多重積分記号の「下に」積分領域を書きたい

例えば体積積分等は三重積分記号の右下に V 等の記号が付きます．

この V 等の積分領域を積分記号の真下中央に置きたい場合があります．もちろん \mathop を使って，手動で記号を置くことも可能ですが，位置合わせを自動化したものに 'multiint.sty' があります．例えば次の三重積分表現は \multiint{3}_{\Omega} の結果です．

$$\iiint\limits_{\Omega} \cdots \tag{13}$$

[56] 'amssymbols.sty' はここにインストールされてないらしい．

8.13.3　閉面積分記号

どうやら $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX にも無い記号らしいのですが，$\oint$ の積分記号が二重になったものを出力するスタイルファイルが ‘Ointint.sty’ です．これを使えば $\oiint$ という積分記号が使えます．\Ointint は picture 環境で楕円を描いているようです．

8.14　黒板書体の数学記号

数学記号の黒板書体というのがあるようです．これは $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX にはあるようですが，通常のフォントを組み合わせたものがあります．ファイルは ‘pmbb-sym.sty’ です．例えば \bbbr が $\mathbb{R}$ になる等です．他の記号を羅列すると，$\mathbb{N}$, $\mathbb{F}$, $\mathbb{H}$, $\mathbb{K}$, $\mathbb{P}$, $\mathbb{1}$, $\mathbb{Q}$, $\mathbb{T}$, $\mathbb{S}$ and $\mathbb{Z}$ になります．

8.15　Commutative diagram を書きたい

これには ‘cd.sty’ と ‘diagram.sty’[57] というスタイルファイルが便利です．‘cd.sty’ (ここにあります) は比較的簡単なものを描くのに便利なようです．サンプルがありますし，使ってないので説明は省略します．

それに比べて ‘diagram.sty’ はかなり混み入ったものも描けるようです．サンプルをそのまま示します．

```
\resetparms
\square[A`B`C`D;\sum_{i=1}^{\infty}`g`h`\Psi^A_k]
```

$$\begin{tikzcd} A \arrow[r, "\sum_{i=1}^{\infty}"] \arrow[d, "g"'] & B \arrow[d, "h"] \\ C \arrow[r, "\Psi^A_k"'] & D \end{tikzcd}$$

もうひとつ LATEX でも plain TEX でも使えるものに ‘diagrams.tex’ (ここにあります) があります．ドキュメントが充実していますから例も使い方も省略しますが，PostScript や TPIC を選択して使うことができます．上で紹介した diagram.sty より使い易いようです．Commutative diagram でなくとも，文字を重ねた矢印や，力学のモーメント等を表現する二重矢印等の利用に便利かもしれません．

8.16　証明における論理の流れ図

なんかよく分かりませんが，‘proof.sty’[58] を使うと良いようです．使ったことがありませんので例等省略します．

8.17　newtheorem のスタイルの変更

“Theorem 4.1” を “Theorem 4-1” のように変えるには，

```
\def\@thmcountersep{-}
```

とするだけです [‘latex.tex’]．

[57] 以前は ‘catmac.sty’ という名前だった．

[58] ‘proof.sty’ はここにインストールされてないらしい．

9.1 箇条書きのスタイルを変更したい

9.1.1 enumerate 環境のラベルを変える

各深さのラベルと表示記号は次の組で記憶されていますから，これを再定義するだけで変更できます [5]．

\labelenumi	\theenumi
\labelenumii	\theenumii
\labelenumiii	\theenumiii
\labelenumiv	\theenumiv
\labelenumv	\theenumv

たとえば一番外側の箇条書きを A), B) のようにしたければ，

```
\def\labelenumi{\theenumi)}
\def\theenumi{\Alph{enumi}}
```

とします．アラビア数字は \arabic{enumi} で指定し，ローマ数字は \roman{enumi} を使います．

9.1.2 itemize 環境のラベルを変える

これも同様で \labelitemi, \labelitemii 等を変更します．例えば次の定義は 2 番目の箇条書きのラベルを「菱形」にします．

```
\def\labelitemii{$\diamond$}
```

9.1.3 enumerate 環境ラベルの記号追加について

enumerate 環境のラベルに記号を付けるものにいくつかあるようです．
ひとつは ‘enumerate.sty’ で，ラベルの番号形式と文字列とをオプションで指定します．

```
\begin{enumerate}[{\bf Problem 1.}]
 \item Solve ...
 \item Explain ...
  \begin{enumerate}[{H{i}nt. a}]
   \item Let ...
   \item Suppose ...
  \end{enumerate}
 \item Calculate ...
  \begin{enumerate}[{H{i}nt. i}]
   \item Let ...
   \item Suppose ...
  \end{enumerate}
\end{enumerate}
```

Problem 1. Solve ...

Problem 2. Explain ...

Hint. a Let ...

Hint. b Suppose ...

Problem 3. Calculate ...

Hint. i Let ...

Hint. ii Suppose ...

オプションの H{i}nt の {i} に注意してください．そうしないと番号形式指示とみなされます．番号形式指示は他に A, I があります．
もうひとつは ‘enumspec.sty’ ですが，こちらは単に数字の前に記号を付けるだけのものです．

```
\begin{enumerate}
 \itemn normal one
 \items specified
  \def\leadchar{\S}
 \itemn normal one
 \items specified
\end{enumerate}
```

1. normal one

$*$2. specified

3. normal one

§4. specified

\itemn が通常の形式で，\items が記号付きになります．

9.2 list 環境の垂直スペースを変更したい

itemize や enumerate 環境は，前後の文章との間に多少広めのスペースが入ります．また \item 毎にも行間が開きます．報告書や本の執筆の場合にはこの方が見易いかもしれませんが，単なるメモや資料では紙の節約の意味も兼ねて，このスペースを小さくしたい場合があります．これは list 環境のスタイルオプションを指定し直せばいいだけですから，

```
\def\enumerate{\ifnum \@enumdepth >3 \@toodeep\else
      \advance\@enumdepth \@ne
      \edef\@enumctr{enum\romannumeral\the\@enumdepth}\list
      {\csname label\@enumctr\endcsname}{\usecounter
        {\@enumctr}\def\makelabel##1{\hss\llap{##1}}%
      \parsep\z@ \itemsep\z@ \topsep\z@}\fi}
%     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
\def\itemize{\ifnum \@itemdepth >3 \@toodeep\else \advance\@itemdepth \@ne
\edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
\list{\csname\@itemitem\endcsname}{\def\makelabel##1{\hss\llap{##1}}%
\parsep\z@ \itemsep\z@ \topsep\z@}\fi}
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
\def\description{\list{}{\labelwidth\z@ \itemindent-\leftmargin
       \let\makelabel\descriptionlabel \parsep\z@ \itemsep\z@ \topsep\z@}}
%                                      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

とするだけで，itemize, enumerate と description 環境のスペーシングが詰まります．このような設定は ‘jeep.sty’[59] というスタイルファイル[60]でもされています．

9.3 description 環境のインデントを揃えたい

description 環境では，見出しの長さによらず固定したインデントが行なわれます．このインデント量を自分で指定できるようにする方法で，文献 [4] に定義されていました．使い方は次のようにします．

```
\begin{namelist}{This length}
\item[first item] This is the first one.
\item[second item] the next one.
\item[last one] finally the last one.
\end{namelist}
```

時　　間	今月今夜 月の出を待って
場　　所	例の場所で
持 参 物	手ぶらでいいよ

この例の見出しには均等割付 (第 13.4 節 (p. 77) 参照) をしてみました．

[59] ‘jeep.sty’ はここにインストールされてないらしい．

[60] もともとの ‘article.sty’ 等がセダン級に相当することを文字って付けられた名称らしい．

9.4 対話を書きたい

これもリスト環境を使って実現できます．リスト環境は他にもさまざまな用途に使える便利なものです．

```
\newenvironment{dialog}{%
  \begin{list}{}{%
    \setlength{\labelwidth}{3cm}
    \setlength{\labelsep}{1cm}
    \setlength{\itemsep}{0pt}
    \setlength{\leftmargin}{3.5cm}
  }%
}{%
  \end{list}}
\def\character#1#2{\def#1{\item[{\bf #2:}]}}
\def\naration#1{\item \par
        \medskip\hspace{-\leftmargin}
        \parbox{0.98\textwidth}{\it #1}
        \smallskip}
```

と定義します．そうしておいて

```
\begin{dialog}
\character{\kame}{亀}
\character{\kuma}{熊}
\naration{長屋の熊公が隣の亀公の家をのぞきこんで}
\kuma 亀公，昨晩の綺麗どころはいってぇー，どうしたってわけだい
\kame なぁにね，昨日も仕事が終わっていつものようにお稲荷さんに
            ちとご挨拶ってね，お参りをしたんだがな
\kuma おう，お稲荷さんってぇのはあのけちの大家んちの裏のかい
\end{dialog}
```

とすればいいようです．実行すると次のようになります．

長屋の熊公が隣の亀公の家をのぞきこんで

熊: 亀公，昨晩の綺麗どころはいってぇー，どうしたってわけだい

亀: なにね，昨日も仕事が終わっていつものようにお稲荷さんにちとご挨拶ってね，お参りをしたんだがな

熊: おう，お稲荷さんってぇのはあのけちの大家んちの裏のかい

9.5 Q & A を書きたい

これは enumerate 環境でできそうですが，2 つの \item 毎に同じ見出し数字にする必要があります．ここでは，上の例のような list 環境による新しい定義によるのではなく，enumerate 環境の簡単な修正でマクロを作成してみました．例えば

```
\def\QandA{\setbox\@tempboxa\hbox{{\bf ~Q.99~:~}}%
  \labelwidth\wd\@tempboxa \leftmargini\labelwidth
  \advance\leftmargini\labelsep \def\labelenumi{\theenumi~:~} \enumerate}
\let\endQandA\endenumerate
\def\Qitem{\@ifnextchar [{\@item}{\@noitemargtrue \@item[\@qitemlabel]}}
 \def\@qitemlabel{{\bf Q.\@itemlabel}}
\def\Aitem{\@ifnextchar [{\@aitem}{\@noitemargtrue \@aitem[\@aitemlabel]}}
 \def\@aitemlabel{{\bf A.\@itemlabel}}
\def\@aitem[#1]{\addtocounter{\@listctr}{-1}\@item[#1]}
```

としておくと

```
\begin{QandA}
\Qitem Why ... How ... ?
\Aitem Because .... In order to ...
\Qitem Then .... ?
\Aitem Yes ...
.....
\end{QandA}
```

Q.1 : どうして○△は□◇になるのか？

A.1 : そんなことも分からないのですか．そんなことだと将来が思いやられます．もう少ししっかりしてくださいよ．

Q.2 : では．．．

A.2 : そうです．．．

ということになります．ただし，深い `list` 環境でのインデントの設定は考慮していません．

図・表・写真はどうしますか

10.1　キャプションを付ける

10.1.1　キャプション見出しや図表番号付けの規則を変更したい

『図 **1**：...』等となるのを『**Fig.** 1. ...』等に変更したい場合があります．本来は `\def\fnum@figure` を変更する必要がありますが，たいていの場合これは

```
\def\fnum@figure{\figurename \thefigure}
```

と定義されているので，この `\figurename` を変更するだけです．つまり

```
\def\figurename{Fig.}
```

とします [‘`j-article.sty`’ etc.]．表の場合も同様です．この文書では

```
\def\figurename{表--}
```

としました．ただし アスキー 日本語 TEX の場合には，直接 `\def\fnum@figure` を定義し直す必要があるかもしれませんので，上の変更で効き目が無い場合には検討してください．

また通常はコロン (:) が付いていますが，このあたりを変更するには `\long\def\@makecaption` の定義部分を変更します．例えばこの文書のようにコロン無しで太字にするには

```
\long\def\@makecaption#1#2{%
   \vskip 10\p@
%   \setbox\@tempboxa\hbox{#1: #2}%        <--- original
   \setbox\@tempboxa\hbox{{\bf#1}\ \ #2}%
   \ifdim \wd\@tempboxa >\hsize
%       #1: #2\par               %        <--- original
      {\bf#1}\ \ #2\par
     \else
      \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
   \fi}
```

のようにします．さらにキャプション内の文字サイズを変更するためには `\long\def\@caption` の定義の中で変更します [‘`latex.tex`’]．

また article スタイルでの図表の番号は，文書の最初から順に振られます．これを例えば節毎にリセットして 1 から始めたい場合には第 8.1.2 節 (p. 37) での設定と同様，\@addtoreset{figure}{section} とすれば図番号が節毎に 1 からになります．表の場合は table というカウンタを同様に設定します．

10.1.2　長いキャプションをインデントしたい

キャプションが短い場合は中央揃え，長い場合は単に複数行にわたります．図表番号と見出しとはそのままで，キャプション本文部だけのインデントを揃えたい場合があります (土木学会関連ファイルがそうです)．次の方法は上の太字の設定をした上でそのような変更をした渡辺さんのアイデアです．

```
\long\def\@makecaption#1#2{%
   \vskip 10\p@
   \setbox\@tempboxa\hbox{{\bf #1}\ \ }%
    \@tempdima\hsize\advance\@tempdima-\wd\@tempboxa
    \setbox\@tempboxa\hbox{{\bf #1}\ \ #2}
   \ifdim \wd\@tempboxa <\hsize
       \hfil {\bf#1}\ \ #2\hfil\par
     \else
       \hbox to\hsize{\hfil {\bf #1}\ \ \parbox[t]\@tempdima{#2}\hfil}%
   \fi}
```

文書部分の幅を判別した上で LaTeX の \parbox を使っています．この例では左右にマージンが入りませんから，キャプションは文書全幅にまでわたることになります．これに対し ‘hangcaption.sty’ (ここにあります) を用いればマージンが適切に設定されます．これは \caption{...} の代わりに \hangcaption{...} とするだけなので例題は省略します． また第 7.10.3 節 (p. 32) でも紹介した稲村氏による定義が ‘hanging.tex’[61]に保存されています．

10.1.3　キャプションの中で使うとエラーが出るコマンドがある

これも \section{...} と同様ですから \protect を用います．第 6.1 節 (p. 23) 参照．

10.1.4　キャプション中に強制改行を入れたい

どうやら \caption の定義中には改行が入れられないようです．これを克服するために \usebox を使ってみました．例えば，

```
\newsavebox{\@parc@ption}
\def\parcaption#1{%
    \sbox{\@parc@ption}{\shortstack[l]{#1}}%
    \setbox\@tempboxa\hbox{\csname fnum@\@captype\endcsname}%
    \@tempdima\columnwidth \advance\@tempdima-\wd\@tempboxa
    \@tempdimb.8\@tempdima %<-- maximum length is set here
    \ifdim\wd\@parc@ption>\@tempdimb \@tempdima\@tempdimb
     \else\@tempdima\wd\@parc@ption\fi
    \sbox{\@tempboxa}{\parbox[t]{\@tempdima}{#1}}%
    \caption{\usebox{\@tempboxa}}}
```

というマクロを定義しておいて，

[61] ‘hanging.tex’ はここにインストールされてないらしい．

```
\begin{figure}
\vspace{?cm}
\parcaption{This is a long caption\\ with line-break.}
\end{figure}
```

という使い方をすれば \parcaption{...} の中では改行が可能となります．

10.2 自分の気に入った位置に図表を入れたい

なかなかどんぴしゃりの位置に図を入れるのは難しく，現実的には諦めた方がいいようですが，いくつか方法はあります．

- オプションできちんと位置を指定する．つまり \begin{figure}[hbtp] のように位置指定をします．
- \begin{figure} の位置をやや早めの位置に動かしてみる．
- フロートの挿入等に関するパラメータを甘くする．これには，topnumber, topfraction 等のパラメータ [4, 5] を定義し直して，同じ頁に入れられる図表の数を増やすと良いようです．各パラメータの定義については参考書を参照してください．この文書でも，

  ```
  \def\textfraction{.08}
  \def\topfraction{.9}
  \def\floatpagefraction{.99}
  \def\dbltopfraction{.9}
  \def\dblfloatpagefraction{.99}
  ```

  としてあります．
- あるいは 'here.sty'[62] を使う．これは強制的に指定場所に図・表を置くスタイルファイルです．通常通りフロートを設定しますが，位置指定を \begin{figure}{H} のように指定します．これで強制的にそこに図が入りますが，頁の境目等の処理は手動で行なうという犠牲を払う必要があります．

10.3 一段組の文書で横に 2 つの図を並べたい

これは渡辺さんに教えてもらったものですが，フロートの中に minipage を入れればいいだけです．つまり，

```
\begin{figure}
 \begin{minipage}[t]{.47\textwidth}
  \vspace{3cm} % Left Figure
  \caption{Figure in the left.}
 \end{minipage}
\hfill
 \begin{minipage}[t]{.47\textwidth}
  \vspace{3cm} % Right Figure
  \caption{Figure in the right.}
 \end{minipage}
\end{figure}
```

とします．図 –2 (p. 29) に出力例があります．

[62] 'here.sty' はここにインストールされてないらしい．

表 –5 枠開け関連スタイルファイルの比較表

機　　能	‘cut.sty’	‘wrapfig.sty’	‘picinpar.sty’
枠の横幅	指定	指定	自動だが狭い
枠の行数	指定	自動 (時々指定)	自動
枠の開始位置	段落先頭	段落先頭	自由指定
枠の左右位置	左右	左右	左右と中央
table 環境の枠	○	‘wrapfloat.sty’ 使用で ○	○
枠横に式 (中央)	○	○	$\cdots$ を使用
枠横に式 (左寄せ)	式番号×	式番号×	式番号×
新パラグラフ	\Par で指示	空行で ○ (JaWaT EX 以外)	空行で △ (インデント無し)
枠横で \verb 使用	×	○	○

10.4　文章に枠あけをして図表を入れたい

通常は minipage 環境を用いることで対処することになっていますが，これはあまり見栄えがよくありません．そのため，‘wrapfig.sty’[63] というスタイルファイルが存在します．

これを表についてもできるようにしたのが ‘wrapfloat.sty’ です．これを用いると，原則として文章の横に空ける図・表の幅を指定しさえすればいいようになっています (時折それだけではうまくいかず，行数を指定する必要があります．例えば ‘epsf.sty’ には対応していないようです)．また当然ながら，頁の変わり目等に対する処置は手動でする必要があります．できるだけ，このような枠空けを使わないに越したことはないのですが，講演概要集等ではスペースの関係で，どうしても使いたい場合が出てくるようです．

使用するには \begin{wrapfigure} に 3 つのオプションを付けて枠を設定し， \end{wrapfigure} で終えた後にパラグラフを始めれば，そのパラグラフが指定した図を巻き込むような形になります． 3 つのオプションは [行数], {l(r)} で左 (右) 側指定, {幅} の順に付けます．行数は省略可です． JaWaTEX の場合，\end{wrapfigure} に続くパラグラフ先頭には半角のスペースを置いた方が無難です．また同じく JaWaTEX の場合には，巻き込む文章中に空行による複数のパラグラフは書けません．

機能	環境
表	wraptable
図	wrapfigure

似たようなものに ‘picinpar.sty’ (ここにあります) というのもあります．これは窓を空けるものです．ただ \ha をトークンとして定義していますので， JaWaTEX では使えないでしょう．

また ‘cut.sty’[64] というのもあります．これは ‘wrapfloat.sty’ とほぼ同じ働きをします． これ以外に ‘floatfig.sty’[65] というのもありますが，使っていないので説明は省略します．付属の例題が満足な結果にならないような気がします．

主な 3 つの簡単な比較を 表 –5 に示しましたが，結論としては ‘wrapfloat.sty’ がいいと思います．

もうひとつ見つかりました． ‘wrapfloat.sty’ よりよさそうです．

[63] ‘wrapfig.sty’ はここにインストールされてないらしい．

[64] ‘cut.sty’ はここにインストールされてないらしい．

[65] ‘floatfig.sty’ はここにインストールされてないらしい．

それは ‘picins.sty’ です．前に紹介したものと似ていますが，枠付きの空間を左右に開け，その中に図を組み込むものです．枠が 4 種類ほどあります．別途日本語で，例題兼マニュアル様のものを ‘picins-j.tex’ に作成してあります．

この例は右に「立体箱枠付き」で出ます．その枠の中で右上に記号 ♡△ が位置します．

縦横の長さを指定するからか，このように巻き込む文章中にパラグラフが変わっても構わないようになっています．さらに，キャプションは常に「図」になりますが \piccaption で指定します．上の例のように枠の外に出すには \piccaptionoutside が必要です．

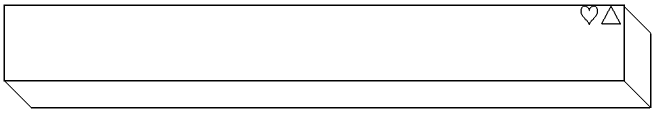


図 –3　枠開けしたところに図

指定の仕方は次のようにします．

```
\parpic(width,height)(x_offset,y_offset)[option][position]{picture}
```

上の例は，

```
\piccaptionoutside
\piccaption{Figure Caption}
\parpic(8.2cm,1cm)[xr][tr]{\picinsymbol}
```

オフセットは枠の指定位置からの移動量， option には上の立体箱の他に，普通の枠 (f) 影付き枠 (s) 破線枠 (d) などがありますが，実はマニュアルがドイツ語なのでこれ以上は分かりません．ただ，左寄せの式環境や verbatim 環境とは両立できないようです．図を枠に入れてしまう \hpic という機能もあるのですが，理解していません．

10.5　複数ページにまたがる表を作りたい

いくつかスタイルファイルがあるようですが，例えば ‘supertab.sty’ (ここにあります) というのがあります．マニュアル・例題も付いています．頁の境を監視して tabular 環境の \end, \begin を出力するマクロです．ひとつ例を示すと次のようなことができます．頁の境と段の移動ごとにそれなりのメッセージが入ります．

```
\tablefirsthead{\hline スタイルファイル名 & \mbox{}\hfill 頁 \\
  \hline\hline}%
\makeatletter%
\tablehead{\hline \multicolumn{2}{l}{\if@firstcolumn%
 前頁より続く \else 左段より続く \fi} \\ \hline スタイルファイル名
  & \mbox{}\hfill 頁 \\ \hline\hline}%
\tabletail{\hline \multicolumn{2}{r}{\if@firstcolumn%
 右段へ続く \else 次頁へ続く \fi} \\ \hline}%
\makeatother%
\tablelasttail{\hline}%
\begin{supertabular}{p{3.5cm}p{1.7cm}}
.....
```

この ‘supertab.sty’ と異なり二段組では使えませんが，頁の縦方向のスペースの制御が優れていそうなものに ‘longtable.sty’ (ここにあります) があります．例題やマニュアルが付属していますからそれを読んでください．入力ファイルは，

```
\begin{longtable}[r]{|l||r|l|}
使用内訳使用使用内訳 & 123,000,000 &
備考備考備考備考備考備考備考備考 \kill % これで幅を
\caption{これが複数の頁に渡る表の例\label{longtab:2}}\\
\hline
使用内訳 & \multicolumn{1}{c|}{金額} & 備考 \\
\hline\hline
\endfirsthead % ここまでが一番最初の表タイトルと項目欄
\caption{複数頁に渡る表 (前頁からの続き)}\\
\hline
使用内訳 & \multicolumn{1}{c|}{金額} & 備考 \\
\hline\hline
\endhead % ここまでが後続する頁の表タイトル付加文字列と項目欄
\hline
 \multicolumn{3}{|l|}{次の頁に続きます} \\
\hline
\endfoot % ここまでが分割された場合の表の最後につけるコメント
\hline\hline
 \multicolumn{3}{|l|}{もうこれが最後の最後} \\
\hline
\endlastfoot % ここまでが最後につけるコメント
県知事 & 12,345,000 & ○△浄水場建設 \\
市長 &  6,789,000 & 市文化センター建設関係 \\
.....
\end{longtable}
```

等とします．デフォールトでは中央揃えになりますが，この例では表を右に寄せており，さらに幅を `\kill` のある行で別途指定しています．これが無指定の場合には通常通り自動的に幅を算定してくれます．ただし最終的なコンパイル時には `\setlongtables` を宣言して，`.aux` ファイルからそれぞれの表のカラム幅を読み込む必要があります．ただし，最後の最後になるまではこの指定ははずしておいた方がいいようです．そうしないと，場合によってはいつまでも幅が決定されないことがあります．

ただし，オリジナルのスタイルファイルのまま用いると，キャプションが "Table 1:" 等となってしまので，変更したい場合には preamble で次のふたつのマクロ定義変更が必要です．

```
\def\LT@c@ption[#1]#2{%
 \LT@mkcaption{{\bf Table \thetable:} #2}% <-----
 \def\@tempa{#1}\ifx\@tempa\empty\else
  {\let\\\space
   \addcontentsline{lot}{table}{\protect\numberline{\thetable}{#1}}}%
 \fi}
\def\LT@@c@ption#1{%
 \LT@mkcaption{{\bf Table \thetable:} #1}% <-----
 {\let\\\space
  \addcontentsline{lot}{table}{\protect\numberline{\thetable}{#1}}}}
```

ここでは単に `\bf` にしてみた例を示してありますが，日本語にする場合にはこの二ヶ所の "Table" を『図－』等に変更するだけです．実例は省略しましたが，上の '`supertab.sty`' と共に例題ファイル '`sp-lgtab.tex`' を作ってあります．

他に '`bigtabular.sty`'[66] 等もありますが，使っていないので説明・例を省略します．

[66] '`bigtabular.sty`' はここにインストールされてないらしい．

10.6 picture 環境で図を描く

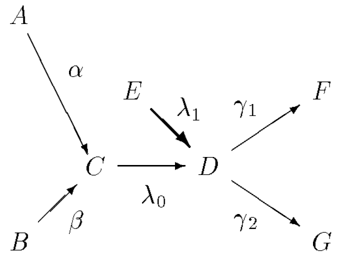
10.6.1 簡単な流れ図

いちいち図化プログラムを使わないでもいいような簡単な流れ図は，標準の picture 環境で十分描けます．しかし，矢印の始点・終点や矢印脇に記号を置くのが結構面倒です．

これを比較的簡単にできるようにしたのが，‘smallgrf.sty’ です．方眼紙を思い浮かべながら点を指定しなければなりませんが，結ぶ矢印や記号は簡単に設定できます．

例えば，右の例はサンプルの真似ですが，次の結果です．

```
\unitlength=0.5mm%
\begin{picture}(80,60)(0,-20)
  \point{P0}( 0, 40){$A$}
  \point{P1}( 0,-20){$B$}
  \point{P2}(20,  0){$C$}
  \arrow{P0}{P2} \midput(5, 5){\cbox{$\alpha$}}
  \arrow{P1}{P2} \midput(5,-5){\cbox{$\beta$}}
  \point{P3}(30,20){$E$}
  \point{P4}(50, 0){$D$}
  \thicklines
  \arrow{P3}{P4} \midput(5,5){\cbox{$\lambda_1$}}
  \thinlines
  \arrow{P2}{P4} \midput(-3,-10){$\lambda_0$}
  \point{Q1}(80, 20){$F$}
  \point{Q2}(80,-20){$G$}
  \arrow{P4}{Q1} \midput(-5, 5){\cbox{$\gamma_1$}}
  \arrow{P4}{Q2} \midput(-5,-5){\cbox{$\gamma_2$}}
\end{picture}
```



10.6.2 フローチャートを書きたい

これも非常に難しいようです．ある程度は ‘Flow.sty’ (ここにあります)　を用いればできるようです．つまり while 文や if 文で記述されたアルゴリズムをフローチャートで表わすことはできますが， goto 文の入ったようなアルゴリズムは記述できないようです．また \ha という長さを用いているので JaWaTEX では使わない方がいいでしょう．サンプルがオリジナルファイルに含まれていますので例は省略します． 他に ‘fchart.tex’[67] というのもありますが，ドキュメントがドイツ語なのとチャートの書式が少し変わっているので使っていません．しかし ‘flowchart.sty’ はいただけません．

スタイルファイルではなく，ある書式で記述した流れ図を LaTeX の picture 環境のコマンドに変換するプログラムがありました．プログラム名はそのまま ‘flow’ です． C で書かれており， UNIX でも MS-DOS でもコンパイルできるようです．流れの分岐やループができる点で上記スタイルファイルより便利です．オリジナルのマニュアルが解りやすくできていますので，使用法等については省略し，例を図 –4に示しておきました．この例中の ¥yen は記号 ¥ を出力するためのものです． \verb が効かないようなのでこうしてしまいました．

10.6.3 木構造図を描きたい

木構造図と言っても縦横の向きやスタイルにいろいろあります．まず縦向きの木構造図には文献 [8] にある磯崎氏の ‘ecltree.sty’ (ここにあります) が使えます．ただし ‘epic.sty’ が必要です．もちろん ‘eepic.sty’ があるともっとよくなります．

[67] ‘fchart.tex’ はここにインストールされてないらしい．

```
Oval
 ¥LaTeX { }に
 よる清書
Box
 ソースファイル
 を編集します
Tilt
 ¥TeX { }で
 コンパイル
Tag
Choice . 有 . 無
 コンパイル
 時エラー
Tag
Left 1
Up 7
Right 1 *
ToTag
Down
Box
 { ¥tt dvi } ファイル
Tilt
 プレビュー
Choice . 有 無 .
 論理や
 レイアウトに
 エラー
Tag
Right
Tilt
 印刷して社会
 に役立てる
Down
```

```
Oval
 出来上がり
ToTag
Left 1
Up 11 *
ToTag
Right
SetTrack line
Box
 ¥LaTeX { } マクロ
 ユーザマクロ
SetTrack none
Up 1
TxtPos [l] [l]
Text
 { ¥yen } documentstyle ¥ { article ¥ }
 { ¥yen } title ¥ { This is a title ¥ }
 { ¥yen } author ¥ { It's ME ¥ }
 { ¥yen } date { { ¥yen } today ¥ }
 { ¥yen } begin ¥ { document ¥ }
 { ¥yen } maketitle
 .....
Down 6
Text
 ←このへんが
 ←やはり一番
 ←面倒な作業
Down 3
Text
 ←プレビューが
 ←大画面でもっ
 ←と鮮明ならと
 ←思うのだが！
```

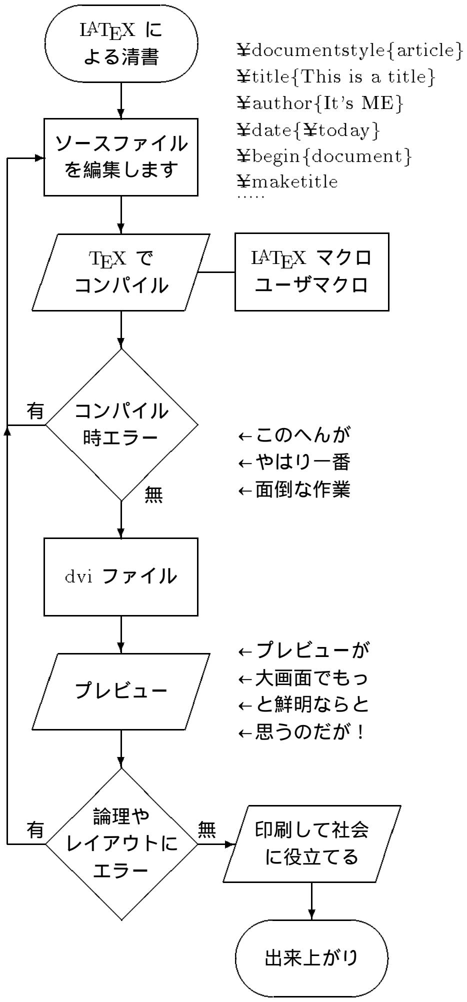


図 –**4** フローチャートのデータと出力

オリジナルファイルに使用法は付属していますが，そこからひとつ抜き出したものが 図 –**5** の例です．ただし，ここでもわざと ‘`eepic.sty`’ は使っていません．このスタイルファイルでは，この例以外にも数種類の図を描けます．

```
\begin{bundle}{xxx}
\chunk{aaa}
\chunk{
        \begin{bundle}{yyy}
        \chunk{bbb}
        \chunk{ddd}
        \end{bundle}}
\chunk{ccc}
\end{bundle}
```

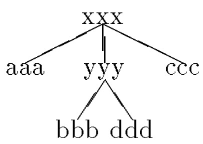


図 –**5** ‘`ecltree.sty`’ を使った木構造図の例

また，横向きのものには ‘`tree.sty`’ というのがあります． 図 –**6** の例は，オリジナルの例を変形

```
\treeroot{Animal}
 {\branch*{Bird}
   {\LEAF{Penguin\\%
             Cardinal}
    \LEAF*{Ostrich}
    \LEAF{Robin}}
  \branch{Mammal}
   {\LEAF{Human Being}
    \branch{Dog}%
      {\LEAF{Sheep Dog}
       \LEAF*{Dachshund}}
    \LEAF{Cat}}}
```

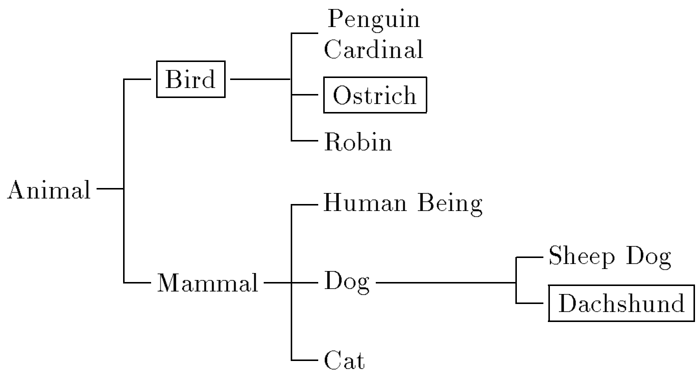


図 –**6** ‘tree.sty’ を使った木構造図の例

したものです．

さらに，ディレクトリ構成図のようなもの用に ‘eclclass.sty’ があります．オリジナル例の言葉を替えてみたのが 図 –**7** です．

```
\begin{classify}{/}
\classf{etc}
\class{\begin{classify}{\fbox{usr}}
       \classf{bin}
       \class{\begin{classify}{\fbox{local}}
              \class{bin}
              \class{lib}
              \class{man}
              \class{lady}
              \end{classify}}
       \classf{global}
       \end{classify}}
\end{classify}
```

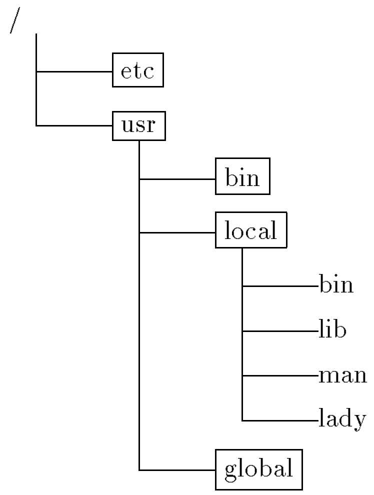


図 –**7** ‘eclclass.sty’ による木構造図の例

10.6.4 棒グラフを描きたいのですが

棒グラフくらいなら通常の LaTeX の範囲で描けそうです．通常の設定のままで ‘bar.sty’ (ここにあります) というスタイルファイルが使えます．基本的な使い方はドイツ語のドキュメントが読めなくても大体分かりますが，そこに書かれていない機能もたくさんあるようなので，別途日本語で[68]，例題兼マニュアル様のものを ‘bargraph.tex’ に作成してあります．

10.7 表の線の太さや属性を変えたい

線の太さは \arrayrulewidth で変更できます．また破線については，文献 [8] にあるものは水平線のみですが任意のカラムに破線を引けます．ここではそれを \dline としました．デフォールトの破線パターンは

[68] ここに含めてしまうとこの文書そのものが string 不足でコンパイルできなくなるので別ファイルになっている．

表 –6 表中の上下の詰まり過ぎを防ぐ

Laplace transforms		
$f(t)$	(p)	no struts
$\delta(t)$	1	optional arg.
$\cos\omega_0 t$	$\frac{p}{p^2+\omega_0^2}$	

Laplace transforms		
$f(t)$	(p)	with struts
$\delta(t)$	1	optional arg.
$\cos\omega_0 t$	$\frac{p}{p^2+\omega_0^2}$	

4pt の長さに 2pt の実線部分を持ったものですが，`\dlinepattern{2pt}{4pt}` で変更できます．

```
\begin{tabular}{|l|l|} \hline
Item A & Contents A \\
 \cline{1-1} \dline{2-2}
Item B & Contents B \\ \hline
\end{tabular}
```

`\arrayrulewidth`	0.4pt (default)
`\dline{...}`[8]	`\cline{...}` と同様

これに対し，鉛直線も破線にするものは ‘`arydshln.sty`’ というスタイルファイルになっており，破線は表や行列の全幅・全高さに及びます．第 8.4.1 節 (p. 38) のように行列を小行列に分割表示する場合などにも便利です． この場合は `\hdashlinewidth` と `\hdashlinegap` とが破線の幅と間隔 (デフォールトは `4pt`) を定義し，

```
\begin{tabular}{|l:l|} \hline\\
Item A & Contents A \\ \hdashline
Item B & Contents B \\ \hline
\end{tabular}
```

:	縦の破線
`\hdashline`	横の破線

のように使います．

10.8 表の上下方向の詰まりすぎを補正したい

これも TEX and TUG NEWS (Vol.2, No.3 July 1993 ‘`ttn2n3.tex`’) にあったもので， Claudio Beccari 氏 (Politecnico di Torino, `beccari@polito.it`) によるものです．

```
\newcommand\setTBstruts{\def\T{\rule{0pt}{2.6ex}}%
                        \def\B{\rule[-1.2ex]{0pt}{0pt}}}
```

としておいて，適当な場所で `\setTBstruts` を宣言しさえすれば，表中で上下のスペースを `\T` および `\B` で調整できます．表 –6 の右は

```
\begin{tabular}[t]{|c|c|l|}
   \hline
   \multicolumn{3}{|c|}{Laplace transforms\T\B}\\\hline
   $f(t)$       \T & $(p)$ \B      & with struts  \\\hline
   $\delta(t)$     & $1$    \T     & optional arg.\\[.5ex]
   $\cos\omega_0t$ & $\frac{p}{p^2+\omega_0^2\B}$ & \\\hline
\end{tabular}
```

の出力です．

表 –7 dcolumn.sty を使った小数点揃え

abc	def	ghi	jkl	mno
d{-1}	d{2}	d{1}	.	,
1.2	1.2	1.2	1.2	1,2
1.23	1.23	1.23	1.23	1,23
1234.5	1234.5	1234.5	1234.5	12,345
123.45	123.45	123.45	123.45	1,234,567
987	987	987	987	987
.4	.4	.4	.4	,4

10.9　小数点を揃えて表にしたい

これについても簡単な方法が文献 [8] にありますが，

ここではスタイルファイル 'dectab.sty' を紹介します．これは自動的に小数点を揃えてくれます．ただ整数の場合には小数点を省略されてしまいます．例を示すと次のようになります．

```
\begin{tabular}{lcc}
Constant  & \dec $-$1.461 & \dec $-$0.724  \\
Constant  & \dec $-$2.    & \dec $-$0.0724 \\
Std Error & \dec  (0.422) & \dec  (0.455)  \\
\end{tabular}
```

Constant	−1.461	−0.724
Constant	−2	−0.0724
Std Error	(0.422)	(0.455)

もうひとつあります．こちらの方が優れています．

それは，表作成の種々の tip を含んだスタイルファイル群として存在する Mittelbach 氏の Mainz File に含まれているものです，'dcolumn.sty' というものです．ただし，見出しは \multicolumn を用いる必要があります． 表 –7 の例は，

```
\newcolumntype{.}{D{.}{.}{-1}}
\newcolumntype{,}{D{,}{,}{2}}
\newcolumntype{d}[1]{D{.}{.}{#1}}
\begin{tabular}{|d{-1}|d{2}|d{1}|.|,|} \hline
  \multicolumn{1}{|c|}{abc} & \multicolumn{1}{c|}{def} &
   \multicolumn{1}{c|}{ghi} & \multicolumn{1}{c|}{jkl} &
    \multicolumn{1}{c|}{mno} \\
 \multicolumn{1}{|c|}{{\tt d\{-1\}}} &
  \multicolumn{1}{c|}{{\tt d\{2\}}} &
   \multicolumn{1}{c|}{{\tt d\{1\}}} & \multicolumn{1}{c|}{.} &
    \multicolumn{1}{c|}{,} \\ \hline
1.2 & 1.2 & 1.2 & 1.2 & 1,2 \\
1.23 & 1.23 & 1.23 & 1.23 & 1,23 \\
1234.5 & 1234.5 & 1234.5 & 1234.5 & 12,345 \\
123.45 & 123.45 & 123.45 & 123.45 & 1,234,567 \\
987 & 987 & 987 & 987 & 987 \\
.4 & .4 & .4 & .4 & ,4 \\
 &&&& \\ \hline
\end{tabular}
```

表 –8 `tabularx.sty` を使った例

item	description	price	remarks
one	This is a pen made from spaghetti using special glues ...	1,000	Eat with care. No smoking please[70]

表 –9 `array.sty` のひとつの例

p{2.1cm}	m{2.4cm}	b{2.1cm}	|<- !{\hspace{1cm}}l
上揃えの parbox になる	中央揃えの parbox になる	下揃えの parbox になる	左に 1cm 余白

の出力です．最初のコンマで揃ってしまうために幅が足りず，第五カラムではエラーが発生します．左から三番目のカラムではエラーは出ませんが，小数点以下が二桁あるのに一桁分しか余裕を取っていませんから，表が詰まってしまいます．小数点以下を負にした第一，第四カラムは少し間が空きすぎます．

あるいは ‘decalign.sty’[69] というのもあるようですが，例は省略します．

10.10 全幅を規定した表の作成

通常は `\begin{tabular*}{12cm}` で作成します．ただ `@{\extracolsep{\fill}}` を適宜入れておかないときれいにはなりません．この置き場所の選択にコツが要ります．

これを簡略にできるようにしたのが ‘tabularx.sty’ です．脚注も含められますし，幅を設定するのに `X` というのが使え，自動的に幅をうまく取ってくれます．例えば， 表 –8 の例は，

```
\begin{tabularx}{.75\textwidth}{|l|X|r|X|} \hline
 item & description & price & remarks \\ \hline
 one  & This is a pen made from spaghetti using special glues ...
 & 1,000 & Eat with care. No smoking please\footnote{%
    Strictly no cigar!} \\ \hline
\end{tabularx}
```

の出力です．ただし `\verb` 等はうまく使えません．

10.11 array, tabular 環境の改善

縦線を入れる従来の `|` は，その線の幅を勘定していなかった等の不都合があるらしく，それを改善したスタイルファイルが存在します．

それは ‘array.sty’ です．従来の `@{...}` もカラムとカラムの間のスペースを殺した上で，間に `...` を挿入していましたが，例えば新しい `!{...}` はカラム間のスペースはそのままにしておいて `...` を挿入することができます． 表 –9 は，

---

[69] ‘decalign.sty’ はここにインストールされてないらしい．

表 –10 array.sty のもうひとつの例

c	>{$}c<{$}	>{\bf}c	c	>{\sc}c
a-b*c	$a-b\times c$	**Column**	Column	COLUMN

```
\begin{tabular}{|p{2.1cm}|m{2.4cm}|b{2.1cm}|
                            !{\hspace{1cm}}l|} \hline
 {\tt p\{2.1cm\}} & {\tt m\{2.4cm\}} & {\tt b\{2.1cm\}} &
  {\tt |<- !\{\symbol{'134}hspace\{1cm\}\}l} \\ \hline
上揃えの parbox になる & 中央揃えの parbox になる & 下揃えの parbox になる &
左に 1cm 余白 \\ \hline
\end{tabular}
```

の結果です．

これに加えて >{...} と <{...} とで，並び指定の前後に宣言が可能となります．例えば，第 8.4.3 節 (p. 40) の 'delarray.sty' の例でも L という配列宣言が使われています．表 –10 は，

```
\begin{tabular}{|c|>{$}c<{$}|>{\bf}c|c|>{\sc}c|} \hline
 c & \mbox{{\tt >\{\$\}c<\{\$\}}} & {\tt >\{\symbol{'134}bf\}c} &
  c & {\tt >\{\symbol{'134}sc\}c} \\ \hline
a-b*c & a-b\times c & Column & Column & Column \\ \hline
\end{tabular}
```

の結果です．マニュアルには >{\setlength{\parindent}{1em}}p といった使い方も示してありました．また第 10.9 節 (p. 59) でも使ったように，\newcolumntype で新しいカラム宣言記号を定義することもできます．この $ を使ったものを数式環境中 array 環境で使うと，ここことは逆に，一時的に数式環境から抜け出すことができます．

10.12 表左上隅の箱に斜めの線を入れたい

スケジュール表等の場合，表の左上隅の欄には斜めの線を入れ，その左下・右上に行と列との意味を列記することがあります．通常 [LA]TEX では任意角度の斜め線フォントが用意されていませんが，その工夫をしてあるスタイルファイルに 'slashbox.sty' があります．これを使うと，次のような表 (例題から直接頂きました) が作れるようです．

月／日 ＼ のりもの	5/31	6/1	6/2	6/3	6/4
自転車	学部長	教務課長	庶務掛長	用度課長	事務長
一輪車	事務部長	学部長	某学科教授	事務部長	学部長秘書
三輪車	某学科教授	事務長	—	—	施設掛
リアカー	設備掛	施設掛	某学科用務員	—	学部長 (運動会)

これは次の出力です．

```
\begin{tabular}{|l||*{5}{c|}}\hline
\backslashbox{のりもの}{月／日}
& \makebox[3em]{5/31} & \makebox[3em]{6/1} & \makebox[3em]{6/2}
& \makebox[3em]{6/3} & \makebox[3em]{6/4} \\ \hline\hline
```

```
  自転車 & 学部長 & 教務課長 & 庶務掛長 & 用度課長 & 事務長 \\\hline
  一輪車 & 事務部長 & 学部長 & 某学科教授 & 事務部長 & 学部長秘書 \\\hline
  三輪車 & 某学科教授 & 事務長 & --- & --- & 施設掛 \\\hline
  リアカー & 設備掛 & 施設掛 & 某学科用務員 & --- & 学部長（運動会）\\\hline
\end{tabular}
```

仕上がり具合を調整する方法は，スタイルファイルに記述があります．また斜めの線は epic.sty, eepic.sty をオプション指定すると綺麗になるという報告もあります．

10.13　図表に副番号を付けたい

第 8.1.1 節 (p. 35) の数式番号と同様，一連の図表を同じ番号のまま副番号で，例えば 図 –**1 (a)** 等の記号でそれぞれを区別したい場合があります．例えば 'subfigure.sty'[71] 等のスタイルファイルがありますが，次のように \caption コマンドを変更することによっても実現できます． これと同様のものが 'subfloat.sty' (ここにあります) でも定義してあります．こちらでは図番号無しの副番号・記号だけのものや，その記号も無いキャプションも可能になっています．使い方は付属していますからここでは省略します．

ここでは，『図のキャプションは下に，表のキャプションは上に』くることを前提としています．

```
\@definecounter{subfloatnumber}
\def\@T@bLe{table}
\def\subfloatcap{-\alph{subfloatnumber}}

\def\subcaption{\@addtoreset{subfloatnumber}{\@captype}%
\ifx\@captype\@T@bLe\relax\else\addtocounter{\@captype}{1}\fi%
\def\the@subfloatnumber{\csname the\@captype\endcsname\subfloatcap}%
\stepcounter{subfloatnumber}%
\edef\@currentlabel{\csname the@subfloatnumber\endcsname}%
\@dblarg{\@subcaption\@captype}}

\long\def\@subcaption#1[#2]#3{\par\addcontentsline{\csname
  ext@#1\endcsname}{#1}{\protect\numberline{\csname
  the#1\endcsname\subfloatcap}{\ignorespaces #2}}\begingroup
    \@parboxrestore
    \normalsize
    \@makecaption{\csname fnum@#1\endcsname\subfloatcap}%
         {\ignorespaces #3}\par
    \endgroup%
     \ifx\@captype\@T@bLe\relax\else\addtocounter{\@captype}{-1}\fi}
```

使い方は

```
\begin{figure}
...
 \subcaption{...} \label{fig:thisone-1}
...
 \subcaption{...} \label{fig:thisone-2}
\caption{Main Caption} \label{fig:thisone}
\end{figure}
```

とします．表の場合も同様です．例を 図 –**2** (p. 29) や第 3.3 節 (p. 11) のサンプル中 (もし出力されていれば) の表に示しました．

[71] 'subfigure.sty' はここにインストールされてないらしい．

10.14 写真環境を作りたい

図や表はフロートと呼ばれる環境に入れることが原則です．つまり，文章中を移動できる箱に入れるわけです．このとき，キャプションはその環境の名前で見出しが決定されます．ここでは写真用の環境を作ってみます．それには，

```
\newcounter{photo}
\def\thephoto{\@arabic\c@photo}
\def\fps@photo{tp}
\def\ftype@photo{4}
\def\ext@photo{lop}
\def\fnum@photo{\photoname \thephoto}
\def\photo{\@float{photo}}
\let\endphoto\end@float
\@namedef{photo*}{\@dblfloat{photo}}
\@namedef{endphoto*}{\end@dblfloat}
\def\photoname{Photo}
```

とすればいいでしょう．4 行目の `\ftype@photo` が ‘4’ になっています．もし，プログラム環境等を既に宣言してあれば，この数字は変更しなければなりません．これは ‘2’ のべき乗になっている必要があります．この設定をしておけば，

```
\begin{photo}
...
\caption{This is a beautiful picture.}
\end{photo}
```

とすればキャプションが ‘**Photo 5:** This is ...’ となります．また ‘`\jobname.lop`’ に写真目次も出力されることになります．もし `report` スタイル文書で写真目次も作る必要がある場合には，

```
\def\listofphotos{\@restonecolfalse
  \if@twocolumn\@restonecoltrue\onecolumn\fi
  \chapter*{\listphotoname\@mkboth{\uppercase{\listphotoname}}%
   {\uppercase{\listphotoname}}}\@starttoc{lop}\if@restonecol
  \twocolumn\fi}
\let\l@photo\l@figure
```

を宣言しておけば，その目次を出力したい箇所に `\listofphotos` の行を挿入するだけで目次ができます．ただ `\listphotoname` を定義しておく必要があり，例えば NTT jTEX の ‘`j-report.sty`’ 的にすると，

```
\def\listphotoname{写真一覧}
```

とでもします．あるいは上の定義中に直接書き込んでも構いません．ただ `report` スタイルでは，図表番号は章毎にリセットするようになっています．これをするためには，

```
\@addtoreset{photo}{chapter}
```

としておきます．あるいは，カウンタの初期宣言のときに

```
\newcounter{photo}[chapter]
```

とします．

こういった新しいフロートの定義を簡単にできるような機能や，フロートの出力形式を箱詰め等に変更する補助ファイルとして ‘`float.sty`’[72] というものがありますが，使っていないので説明は省略します．

[72] ‘`float.sty`’ はここにインストールされてないらしい．

10.15 図や絵を読み込んで直接はめ込む

報告書や論文を作成していて一番厄介なもののひとつが図の切り貼りです．できればこれも自動的にやって欲しいところです．

もともと LaTeX には絵を描くための『picture 環境』がありますが，この環境では図のための数少ないフォントを用いた図形を描くだけなので，半径や傾き等に強い制限があります．したがって，簡単な地図程度ならば比較的容易に図化できますが，複雑な曲線を含む図やグラフ，細かい絵・写真等をこの環境内で処理することは困難です．これを解決するために，もっときめ細かい図用のフォントを作成して対処しようとする努力もされているようです (多分 xypic というのがそれかな？)．

あるいは，既存のフォントだけを使うものの，図のためのきめ細かいマクロを作成したものもあります．これは PICTEX と呼ばれるものですが，説明は割愛します．これを用いるには一般に BigTEX が必要な上にコンパイル時間がかかります．マニュアルは有料です．

もうひとつの打開策は，ドライバやデバイスに依存したコードを用いる方法です．この方法のほとんどでは LaTeX の picture 環境の制限を無くすことができますが，例えば複数の人間が異なる環境で分担執筆している場合等に整合性の問題が発生してしまいます．また，機種環境によっては利用できない可能性も出てきます．しかし，データの作成が容易だったり，コンパイルが非常に速くなる等のメリットがあります．

ここでは，比較的容易でしかも広範な環境で利用できる方法について概説してみます．

10.15.1 LaTeX の枠組みの範囲内でできること

前述のようにこの方法には強い制限があるものの，特別な措置を必要としないメリットがあります．

1. LaTeX の picture 環境を使う: 方眼紙の下書きを元にその座標値を丁寧に入力するか，支援ユーティリティを利用します．この方式のファイル出力をするソフトには，gnuplot[73] や xfig[74]，texdraw[75] 等があります．出力ファイルを文書中に直接含めるか \input で読み込みます．

2. PICTEX を使う : 命令の説明は割愛しますが，preamble で

```
\documentstyle{article}
\input{prepictex.tex}
\input{pictex.tex}
\input{postpictex.tex}
....
```

   といった読込が必要です．xfig や後述の qfig 等による出力形式にも含まれています．

3. ‘bitmap.sty’ を使う : 図やスキャナで読み込んだ写真等を何等かの方法でいわゆる X bitmap file format (xbm) 形式に変換しさえすれば，スタイルファイル ‘bitmap.sty’ を用いて文書中に組み込めます．ただし複雑な図形は TEX の能力の限界のためにコンパイルできない場合があります．

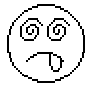
[73] UNIX, MS-DOS 用共にある．

[74] akiu:ftp/pub/X11R5/contrib-fixes/ の中．UNIX 用．ファイル出力するためには別途 transfig も必要．

[75] UNIX 用．

‘bitmap.sty’ を使って図を読み込むには

```
\bitmapinput[pos]{dpi}{image}
```



のようにします． pos は b, c, t のいずれかで指定する上下方向の位置設定． dpi は絵の解像度です．右の絵は xalive というユーティリティーのアイコンのひとつです．

4. METAFONT を使う: 図そのものをひとつのフォントとして METAFONT で作成してしまう方法も無いわけではありません．これには ‘drawing.tex’[76] を真似することができますが，当然 METAFONT がインストールされている必要があります．詳細についてはドキュメントを参照してください．

10.15.2 デバイス・ドライバ依存の方法

本質的にはデバイス (プリンタ) 依存の方法ですが，比較的多くのプリンタで解釈可能なコードを出力するドライバを用いれば可能な方法のことを，「ドライバ依存」と呼んでいます．デバイス依存の方法の代表はいわゆる PostScript[77] プリンタや LIPS III の利用でしょう．

1. LIPS III の利用: この場合，例えば Graph-P に付属しているプリンタドライバとマクロを用いれば，文中に Graph-P[78]，および Ngraph[79]の「図」を挿入できます．

2. PostScript の利用: PostScript プリンタあるいはそのエミュレータ GhostScript が利用できる環境では，数多くのユーティリティーにより作成した図・絵・写真を ‘epsf.sty’ あるいは ‘epsbox.sty’ を使って直接の読込が可能です．前者は NTT jTeX 用，後者は アスキー 日本語 TeX 用です．ただし，LaTeX に読み込むためには EPSF (Encapsulated PostScript File) 形式である必要[80]があります．free の図化ソフトには gnuplot や Graph-P， Ngraph 等があります．また絵を描くものには xfig，ssdraw[81]， idraw[82] や tgif+[83] 等があります． ‘epsf.sty’ (ここにあります) の使い方は，基本的に

```
\epsfile{file=image.ps,scale=0.5}
```

のように使います．詳細はスタイルファイル付属のサンプルを見てください． ‘epsbox.sty’[84]も同様で

```
\EPSfile{file=image.ps,scale=0.5}
```

のように使います．詳細はスタイルファイル付属のサンプルを見てください．

3. TPIC の利用: この図化言語は比較的多くのデバイスドライバがサポートしているもので，直接 TPIC の規則で記述したものをプログラムで LaTeX 用の \special に変換したり， LaTeX の \special を直

[76] ‘drawing.tex’ はここにインストールされてないらしい．

[77] PostScript は Adobe Systems Inc. の商標．

[78] akiu:ftp/pub/msdos/gp/ の中． MS-DOS 用．

[79] akiu:ftp/pub/msdos/PC98/Ngraph/ の中． MS-DOS 用．

[80] ファイルの始めの方に %%BoundingBox:... という行が無いと駄目．この形式ではない PostScript ファイルの場合は bbfig 等のユーティリティを通して変換する．

[81] akiu:ftp/pub/msdos/misc/SSDRAW/ の中． MS-DOS 用．

[82] UNIX 用．日本語版は kdraw．

[83] UNIX 用．

[84] ‘epsbox.sty’ はここにインストールされてないらしい．

接記述することもできます．あるいは ‘epic.sty’, ‘eepic.sty’ の 2 つ[85]のスタイルファイルを用いれば，比較的簡単な記述で絵や図を表現できます．これが利用できるデバイスドライバは多種存在するらしいですが，パーソナルコンピュータ用で例を挙げれば dviout/prt (Version 2.37 以降， NTT JTEX 用で書体倶楽部フォントを利用する場合は Version 2.38 以降のパッチ版) がそれに当たります．図作成のためには，例えば Ngraph の出力を gra2tpic あるいは gra2eepi という変換プログラム[86]に通せば，前者は TPIC コードを，後者は eepic 対応コードを出力します． dviout/prt (Version 2.39) ではサポートする新しい TPIC コマンドが追加され，その中に文字の回転が含まれました．これによって図の縦軸タイトルを横向きに出力することができますが，上の gra2eepi はそれをサポートしています．また絵作成のためには ssdraw や， eepic 対応なら xfig や，第一筆者作成の qfig[87] 等が利用できます．

‘epic.sty’ と ‘eepic.sty’ について補足しますが，前者は新たにフォント作成等をせずに picture 環境を拡張するためのスタイルファイルで，後者はそれをさらに TPIC コードを含んだ \special に変換するものです．利用する場合には，

```
\documentstyle[epic,eepic,...]{j-article}
```

のような順番に並べる必要があります．またドライバが TPIC をサポートしていない場合には， ‘eepic.sty’ のエミュレーション[88]が利用できます．この場合はコンパイルに時間がかかり一部の機能が損なわれますが，ドライバ依存の無い出力が得られます．つまり TPIC で直接コーディングするより， ‘eepic.sty’ 対応の方が汎用性がありそうです．

これ以外にも種々の方法があるようですが，ここでは割愛します．

‘epic.sty’, ‘eepic.sty’ 共にインストールされていますので， Ngraph のデモのひとつを gra2eepi で変換した図[89]と， qfig で描いた漫画と絵[90]を示しました (図–8, 9)．何か出力できた場合には，そのドライバは TPIC を理解できるものだということになります．

10.16 eepic.sty の拡張は

通常の eepic.sty では \shade の濃度を変化できなかったり，枠線無しの \shade にはできません．これに対応する拡張命令を ‘eepicsup.sty’ (ここにあります) で定義してみました． ‘eepic.sty’ の単なる変更ですから，

```
\documentstyle[epic,eepic,eepicsup,...]{article}
```

の順にオプション指定する必要があります．

濃度は \shade にオプションを付けることによって対処します．オプション無しのデフォルトが通常の場合，つまり \shade[0.5] に相当しますが，このオプション値を 0.1 から 0.9 までで指定できます．

---

85 オリジナルのバグ修正版が akiu:ftp/pub/tex/latex-styles/isozaki_collections/latex-style-misc.tar.Z にある．

86 いずれも akiu:ftp/pub/msdos/PC98/Ngraph の中で， gra2tpic は ‘gra2tp??.lzh’ の方に， gra2eepi は ‘gra2eepi.lzh’ にある．この 2 つソフトには，縦書き等の変換に若干の違いがある．

87 akiu:ftp/pub/msdos/PC98/Ngraph/qfig????.lzh の中． MS-DOS 用． IBM 互換機用のバージョンもある．

88 ‘eepicemu.sty’ や ‘ecleepic.sty’ がそれに相当する．前者は ‘eepic.sty’ と共に配布されており，後者は他の eclware と共に配布されている．

89 Ngraph で挿入した文字列で，漢字・英字混じりや上下添え字付きになっている場合には，それぞれを別々の文字列として分解してファイルに出力してしまうため，それぞれが別々の \put で配置されてしまう．最終的に LaTeX に読み込む積もりなら，漢字のみ，英字のみで位置合わせだけをやっておいた方が簡単．数式も同様で，ギリシャ文字等も漢字を使わずに適当な英字文字列だけで位置決めだけをした方がいい．

90 qfig で数式を配置する場合も，漢字等を使わずに適当な英字文字列だけで位置決めをやっておいた方が後が簡単．

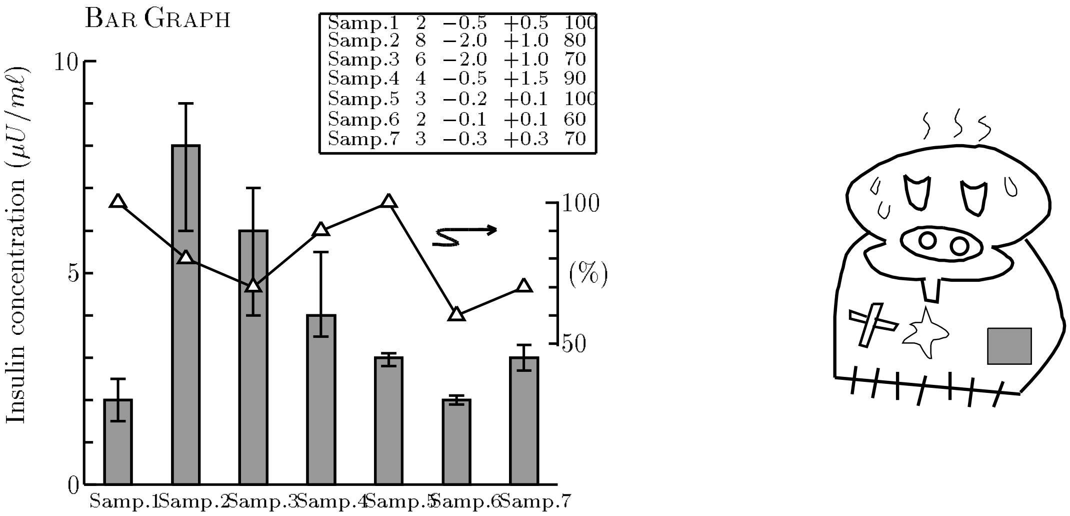


図 –8 `Ngraph`, `qfig` 出力を `TPIC` に変換した図と絵の例

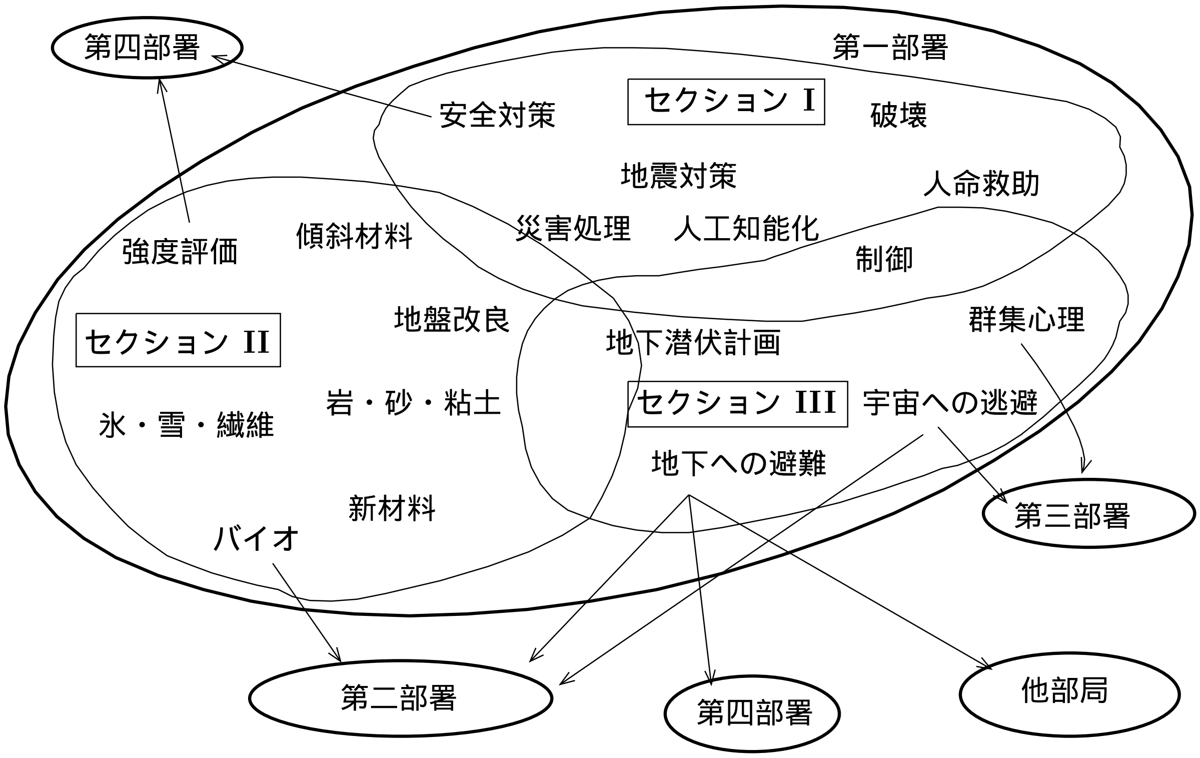
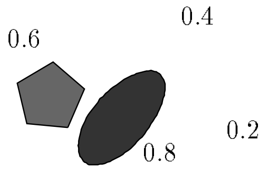


図 –9 中身の無い報告書を粉飾するのに `qfig` を利用した例



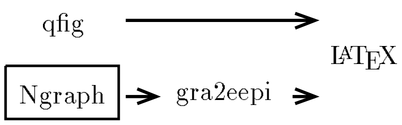

```
\shade[0.2]\put(320,45){\circle{32}}
\shade[0.4]\put(274,71){\ellipse{60}{28}}
\shade[0.6]\path (240,61)(256,53)(......
```

さらに，網掛けの輪郭については `\shadingwithoutoutline` で輪郭を無くせ，`\outlinedshading` で元に戻せるようにしました．

dviout/prt Ver.2.39 以降のサポート

```
\special{pbmfile=earth.pbm hsize=70mm}
\special{pbmfile=earth.pbm vsize=7cm}
\special{pbmfile=earth.pbm vscale=1.2}
  .....
\special{giffile=earth hsize=6cm}
\special{giffile=earth scale=1.5}
\special{giffile=earth hscale=2}
  .....etc.
```

写真 –**1** styleuse.gif (スキャナで読み込みました)




```
\shadingwithoutoutline
\shade[0.8]\put(290,54){\ellipse{52}{28}}
\outlinedshading
\whiten\put(294,50){\ellipse{52}{28}}
\put(284,46){{\xpt\rm qfig}}
```

10.17 写真を読み込んではめ込む

MS-DOS の dviout/prt (Version 2.39) では Portable Bitmap と monochrome GIF 形式の写真を直接読み込めるようです．後者の例を 写真 –**1** に示してみました．新聞の写真と同じ様なディザパターンをかけてあります．他のドライバでどうなるのかは知りません．

```
\begin{photo}
  \special{giffile=styleuse hsize=54.9mm} \vspace*{74.9mm}
    \caption{.....}\label{photo:nn}
\end{photo}
```

10.18 フロートが多過ぎて警告が出る

フロートは数頁分をメモリー内に維持しつつ処理していきます．ですからあまりにも多いフロートがあるとメモリー制限の関係で警告表示になってしまいます．これを回避するひとつの方法は， \clearpage 等で確定できるフロートは確定してしまうことです．ただ，一般にはこれは難しいです．‘morefloats.sty’ (ここにあります) はその制限数を倍増しますが，使ったことがないので紹介だけにします．オプションに並べるだけでいいようです．

箱の中に入れる

11.1　パラグラフを枠の中に入れる

単に minipage 環境を \fbox で囲めばいいようなものですが， verbatim 環境を枠で囲めない等の難点もあります．

ひとつは ‘boxit.sty’ というスタイルファイルで実現可能です．ただし頁をまたぐような箱は第 11.10 節 (p. 71) を参照してください．箱の幅は文字列幅か文書幅かのいずれかにできます．

ここにもありました．単に \begin{boxit} あるいは \begin{boxit*} から始まる部分を箱に詰めます．前者は文字列幅，後者は文書幅の箱を生成します．ただし箱中文章の前後には \vskip .5em 程度の空白が必要でしょう．

また minipage の中の文章はデフォールトではパラグラフ開始のインデントがありません．ですから，自分で \parindent を指定する必要があることに留意してください．

11.2　minipage のまわりに枠をつけたい

最も簡単な方法は minipage を \fbox{...} の中に入れてしまうことですが，これだと， minipage 環境の中で verbatim 環境をそのままは使えないようです．

ひとつの方法は ‘boxedminipage.sty’ を使うことです．通常の minipage と使い方は全く同じで，環境名を boxedminipage にするだけです．

この部分は普通の minipage 環境ですから \verb も使えますが， \fbox では囲めません．これに対して右に示したのが boxedminipage 環境で， verbatim 環境も使えています．

枠で囲った中の verbatim 環境が

```
\begin{boxedminipage}{.45\textwidth}
...
\end{boxedminipage}
```

のように使えます．

11.3　角の丸い枠で文字を囲む

通常は真四角でしか囲めませんが，角を丸めるには ‘loval.sty’ が便利です．同じ作者による ‘oval.sty’ というのもありますが，後者を使うためには新たに円フォントをメタフォントで作成する必要があります．使い方は簡単で，文字を \ovalbox{...} で囲むだけですが，この文書のように そのまま使う と警告が出ているはずです．これは楕円が小さすぎるからのようです．箱の輪郭と文字との間隔を \fboxsep で指定できますが，この大きさのデフォールトが 3pt になっており，これを 5pt 程度にすると 警告が出なく なります．ただ警告が出ても特に問題はありません． 第 11.11 節 (p. 71) の ‘fancybox.sty’ の方がよくできています．

11.4　パラグラフに角の丸い枠を付けたい

これは アスキー 日本語 TEX の場合は screen 環境を使います． NTT JTEX の場合は ‘ascmac.sty’[91] をコピーして使えば同じことができるようですが， boxnote 環境の部分を削除するか，それに対応したフォントをインストールする必要があるようです． ついでですが，この ‘ascmac.sty’ の網掛けとスクリーンマクロを，

[91] ‘ascmac.sty’ はここにインストールされてないらしい．

ドライバ jdvi2kps3.0 用に拡張したものが 'meipsmac.sty'[92] です[93]．他にもプリンタ依存の拡張がいくつかなされています． kusumi@isl.mei.co.jp 氏によるものですが，ドキュメントが丁寧なので，このプリンタが使えないユーザにとっても 'ascmac.sty' の使用マニュアルとして使えます．いずれも使ってないので説明は省略します．

11.5 タイトル付きの箱はできませんか

箱の上辺にタイトルを付けたい場合もあるでしょう．これは Network News (comp.text.tex) に出たものの中に，plain TEX 用に定義されて LATEX でも使えるものがありました．'tboxit.sty' としてみましたが，著者による正式名ではありません．

```
\tboxit{Title}{\vbox{\hsize7.5cm
This is text inside titled box.
This is text inside titled box.
This is text inside titled box.}}
```

これがタイトル

オリジナルのサンプルを真似すると，このようになります．オリジナルのサンプルを真似すると，このようになります．オリジナルのサンプルを真似すると，このようになります．

11.6 図表の部分 (フロート) を枠で囲みたい

一応，フロート部は本文との間にそれなりのスペースが挿入されていますが，枠で明確にしたい場合もあります．

これは 'bigbox.sty' を使えば簡単です．フロート環境内対象部分を \begin{bigbox} と\end{bigbox} とで囲むだけです．試しに 図–**2** (p. 29) と 表–**11** (p. 78) とを囲んでみました．

11.7 影付きの枠で囲みたい

これは箱の右と下に影を付けるものですが，「測量学」の教科書によると地図等で立体感を出すためには，空から見た状態で左上からの斜めの光による影を付けるのが有効とされています．ひとつは 'shadow.sty' を使えば簡単にできます．文章あるいはパラグラフ全体のいずれも可能です．

線の太さや影の大きさを 3 つのパラメータで指定できます．
中身を \shabox{...} の中に入れるだけです．ここでは右にデフォルト値を示しておきます．右のように文字列を縦に並べる簡単な方法は，

```
\shabox{\shortstack[l]{%
 This is the first line \\
    then the \verb+\string\second+ \\
  Finally the last one }}
```

\sboxsep = 10pt
\sdim = 4pt
\sboxrule = .4pt

のようにします．

あるいは，第 11.11 節 (p. 71) の 'fancybox.sty' の方がよくできています．

[92] 'meipsmac.sty' はここにインストールされてないらしい．

[93] meipsmac1.0c.tar.Z というファイル中．

11.8 網掛けをしたい

網掛けは アスキー 日本語 TEX なら第 11.4 節 (p. 69) の 'ascmac.sty' を使えばできるようです． NTT jTEX の場合等に，それを利用する以外の方法としては，第 10.15 節 (p. 64) で説明した TPIC を利用するものが考えられます．第 11.3 節 (p. 69) の 'loval.sty' をほとんどそのまま流用した 'shadedbx.sty' に，その例を示しました．使い方は簡単で，文字を \shadedbox{...} で 囲むだけ です．もちろん TPIC を解釈できるドライバでないと正しく出ません．

他にドライバ依存の無いバージョンもありました．これは 'shadebox.sty' を使う方法です．ただコンパイルに時間がかかり .dvi ファイルが非常に大きくなります．例えば \shade{This is a shaded box.} とすれば This is a shaded box. と出力されます．網の濃度はスタイルファイル中で，プリンタの解像度に合わせて変更する必要があるようです．

11.9 斜めの文字を文章に重ねたい

例えば，機密書類の各頁の文章に赤で "SECRET" と斜めに重ねて印刷するようなことをイメージしています．このようなことは，いわゆる PStricks で何とでもなるらしいのですが，ドライバの dviout/prt (Version 2.39) で， TPIC の拡張機能としての「文字の回転」を提案していますので，それを利用する方法を考えました．これも第 11.3 節 (p. 69) の 'loval.sty' をほとんどそのまま流用した 'tpicrot.sty' で，その例を示しました．

使い方は簡単ですが，内部での計算に 'eclarith.sty' を使っています． (計算そのものは if だらけの目茶苦茶ですが) 3 種類の書き方を定義し， 図 –**10** ~ **12** で試してみました．回転角は反時計回りに度の単位で指定します．

11.10 頁を越えた枠で囲む

文書中に参考程度のコラムを入れたい場合があるかもしれません．これは磯崎氏の 'eclbkbox.sty' (ここにあります) を用います．オプションとして，行番号も付けられます．実行例はスタイルファイルと共に配布されているはずですから省略しますが， breakbox 環境にある部分がすべて枠に含まれ，頁を越えられるようになっています．行番号は \bkcounttrue, \bkcountfalse で On/Off します．

11.11 頁全体や list 環境を箱では囲めませんか

頁全体を囲むのは元々の仕様ではないですから， output ルーティンの変更が必要なようです．ひとつには 'pageframe.sty' (ここにあります) のようなものがあります．これは頁の外に位置合わせのためのクロップマークや，文書中にグリッドを描くものですから，作成中の参考にするものでしょう．単にクロップマークだけなら 'cropmark.sty'[94] というのもあるようです．

これに対し， 'fancybox.sty' というものもあり，これだとそのような頁を囲む箱だけでなく，この節で挙げたような箱詰めをはじめ，二重線による箱等も可能にしています．これはスライドあるいはＯＨＰフィルム原稿作成用の 'seminar.sty' に付属のものですが，単体でも入手可能です．さらに， list 環境等の箱詰めのためのマクロも用意されています．

例えば

[94] 'cropmark.sty' はここにインストールされてないらしい．

```
\begin{center} CONCERT\\
{\sc Tpic Symphony Orchestra}\\
May 16, 1993, 7:00 pm \\
at TPIC Orchestra Hall \\
{\LARGE\bf\Rotatebox*[20]{Cancelled}}
All \LaTeX{} Programs
\end{center}
```

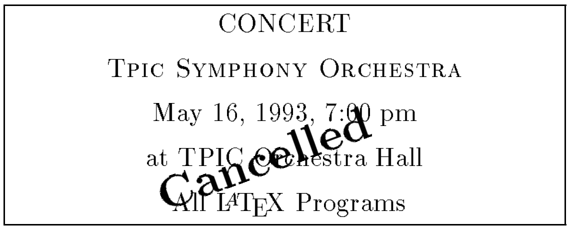


図 –**10** 完全に重なる場合

```
This is a box. This is a box.
This is a box. This is a box.
This is
{\Large\bf\rotatebox[20]{20 degrees}}
a box.
This is a box. This is a box.
This is a box.
```

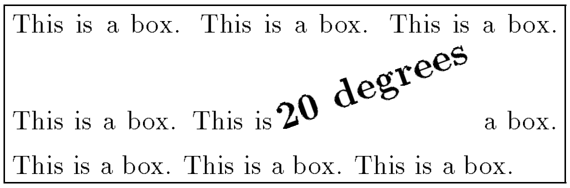


図 –**11** 全く重ならない場合

```
This is a box. This is a box.
This is a box. This is a box.
This is
{\Large\bf\Rotatebox[20]{20 degrees}}
a box.
This is a box. This is a box.
This is a box. This is a box.
```

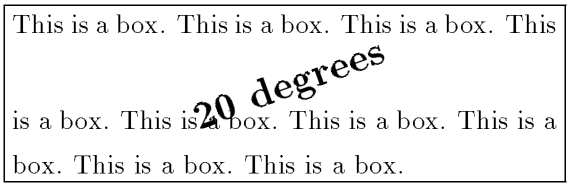
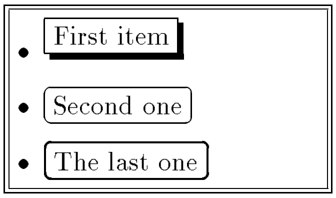


図 –**12** 縦方向だけ重ならない場合

```
\doublebox{\begin{minipage}{.25\textwidth}
\begin{Bitemize}
\item \shadowsize=2pt\shadowbox{First item}
\item \cornersize{.2}\ovalbox{Second one}
\item \Ovalbox{The last one}
\end{Bitemize}
\end{minipage}}
```



なお，巻末の索引頁は

```
\fancypage{\fboxsep=2em\cornersize{2}\ovalbox}{}
```

で囲んであります．その他，箱に関する種々の解説があり，機能の中身や限界を理解するのに便利かもしれません．このファイルで定義されている環境は他に `Bcenter`, `Bflushleft(right)`, `Bitemize`, `Benumerate`, `Bdescription`, `\boxput` と各種 `verbatim` 環境に頁全体の箱詰めです．詳細はマニュアルをご覧ください．

文献引用・参照と索引

12.1 文献引用番号を [1),2)] のようにして肩に乗せたい

`\@cite` を再定義します [‘`latex.tex`’]．通常は

```
\def\@cite#1#2{[{#1\if@tempswa , #2\fi}]}
```

となっているのを

```
\def\@cite#1#2{$^{\mbox{\scriptsize {#1\if@tempswa , #2\fi})}}$}
```

のようにするだけです．あるいはスタイルファイル ‘overcite.sty’[95] を使います．これを使うと，複数の引用をした場合の並べ替えもしてくれます．使い方は通常通りですから割愛します．

12.2 文献リストの番号やタイトルを変更したい

これには thebibliography 環境を書き換えます [‘j-article.sty’ 等]． \def\thebibliography の

```
 {\refname}{\refname}}\list
 {[\arabic{enumi}]}{\settowidth\labelwidth{[#1]}\leftmargin\labelwidth
% ^^^^^^^^^^^^^^^^                          ^^^^
 \advance\leftmargin\labelsep
```

の部分を

```
 {(\arabic{enumi})}{\settowidth\labelwidth{(#1)}\leftmargin\labelwidth
```

とすれば，リストの番号が (1), (2) のようになります．現時点での最新 (Mar. 20, 1992) ‘article.sty’ 等の定義[96]では，

```
\def\@biblabel#1{(#1)}
```

とするだけで同じことができるようになっています．

また

```
\def\refname{文献}
```

とすれば，リストタイトルを「参考文献」から「文献」に変更できます．ただし，これは NTT jTEX あるいは欧文の場合の設定で， アスキー 日本語 TEX の場合は， \thebibliography の定義部分を明示的に書き変える必要があるかもしれません．この \refname は NTT jTEX の ‘j-*.sty’ と欧文の ‘article.sty’ とでは使われていますが，欧文の ‘report.sty’ と ‘book.sty’ とでは \bibname になっていますので注意してください．

12.3 複数引用のときに並べ替えをしてほしい

\cite で複数の文献を引用する場合，例えば \cite{rakuraku,koho,jlamport,isozaki,total} は通常は [4, 2, 5, 8, 6] となります．これを [2,4-6,8] のように，ソートした上で続き番号の場合には途中を割愛してダッシュで結ぶようにしたい場合があります．

この場合には ‘cite.sty’ を用いればいいようです．使い方は通常通りです．上の例は確かに [2,4–6,8] となります．

---

[95] ‘overcite.sty’ はここにインストールされてないらしい．

[96] この最新版では enumi が enumiv になっているため，新しいスタイルファイルによる変更を古いスタイルファイルと共に用いると，すべての引用番号が 0 になるので注意が必要．この enumiv はその部分に全部で 5 箇所ある．

12.4　章あるいは節毎の参考文献リスト

本を書く場合等のような，特別な場合にはこのような必要があるかもしれません．そのためのスタイルファイルとして `chapterbib.sty'[97] というものがあります． あるいは `bibunits.sty'[98] もありますが，これはテンポラリファイル名が MS-DOS では使えない形式になっていますので修正の必要があります．前者より後者の方が汎用的に使えそうですが，バグがあるようなので，ここでは詳細については触れません．

さらに， include されるファイル毎に異なる参考文献スタイルが使えるように，複数の \bibliography を許容するための補助ファイル `bibperinclude.sty'[99] もあります．使ったことが無いので説明は省略します．

12.5　文献データベースから直接文献リストを作りたい

別途文献データベースを作成しておいて (データベースソフトそのものは自作でも市販のものでも構いませんが，そのファイル出力形式をある特定の規則に従うようにしておけば便利です．) そこから文献リストを作成する方法もあります．

ひとつは，データベースに直接，第 2.4.8 節 (p. 7) にあるようなリストを出力させる方法です．

もうひとつは，そのデータベースを，例えば `mybiblio.bib' というファイルに

```
@ARTICLE{doboku1,
  author={T. Doboku and S. Kensetsu and U. Kankyo},
  title={A beam theory with extension and shear deformation
          of {T}imoshenko type in finite displacements},
  journal=seeejsce,
  volume=1,
  number=99,
  pages="103s--113s",
  year=1984,
}
@INCOLLECTION{nanashi,
  author={Tamen-Ressan, S. and M. Nanashi},
  title={Elastic solids with microdefects},
  booktitle={Micromechanics and Inhomogeneity:
                    The Toshio Mura Anniversary Volume},
  editor={G. J. Weng and M. Taya and H. Ab{\'e}},
  pages="297--320",
  year=1990,
  publisher={Springer-Verlag},
  address={New York},
}
@BOOK{timo,
  author={S. P. Timoshenko and J. M. Gere},
  title={Theory of Elastic Stability},
  edition = {Second},
  publisher={McGraw-Hill},
  address = {New York},
  year=1961,
}
```

[97] `chapterbib.sty' はここにインストールされてないらしい．

[98] `bibunits.sty' はここにインストールされてないらしい．

[99] `bibperinclude.sty' はここにインストールされてないらしい．

といったような形式で作成しておく (あるいはデータベースソフトから出力する) ものです． and や論文タイトルの大文字指定等に注意が必要です．ただしこのファイルを使うためには BibTeX というソフトが必要です．

後者が優れているのは，次のようなことができるからです．

- データベースはひとつあればよく，コンパイルする文書の .aux ファイルから必要な文献だけを捜してリストにする．
- リストの書式，例えば本なのか論文なのか，また，タイトル等と論文名の順番や姓名の順等を設定したスタイルファイル .bst を準備しておくことにより，同じデータベースから，異なる書式を要求する異なる論文集への投稿原稿を簡単に作成できる．
- 本文に現われる順番にリストを作成できる (土木学会論文集の書式)．
- 上の例の seeejsce のような論文名の略号が使える．

詳細については種々の文献やマニュアルを参照してください．上のようなデータベースファイルがあれば，

```
\bibliography{mybiblio}
\bibliographystyle{jsce}
```

の 2 行を本文 (例えば 'mydoc.tex') の適切な場所に置き，

```
latex mydoc ↵
bibtex mydoc ↵
latex mydoc ↵
latex mydoc ↵
```

の順でコンパイルをすれば出来上がります．ただし，上の例で \bibliographystyle に指定した 'jsce.bst' という BibTeX 用スタイルファイルは，標準の 'unsrt.bst' を土木学会論文集用に手探りで変更したもので，第 4.5 節 (p. 19) で紹介した jsce.sty パッケージに入っています．このデータベースを元に BibTeX で処理したファイル 'mydoc.bbl' を元に文献リストを作ると

参考文献

[1] Doboku, T., Kensetsu, S., and Kankyo, U. : A beam theory with extension and shear deformation of Timoshenko type in finite displacements. *Structural Engineering /Earthquake Engineering, JSCE*, Vol.1, No.99, pp.103s–113s, 1984.

[2] Tamen-Ressan, S. and Nanashi, M. : Elastic solids with microdefects. *Micromechanics and Inhomogeneity: The Toshio Mura Anniversary Volume*, (Ed. by G. J. Weng, M. Taya, and H. Abé), pp. 297–320. Springer-Verlag, New York, 1990.

[3] Timoshenko, S. P. and Gere, J. M. : *Theory of Elastic Stability.* McGraw-Hill, New York, second edition, 1961.

となります．著者イニシャル (last name first の規則) や編集者等の順番や書式が土木学会論文集用に変更されています．

12.6 Chicago スタイルの文献表示

論文などの執筆要領を書いたものに Chicago Manual of Style といったスタイル[100]があります．これに従った文献引用およびリスティングのスタイルを設定したスタイルファイルがあります．‘chicago.sty’ (ここにあります) がそれです．文献リストのための BIBTEX 用のスタイルファイルの方は ‘chicago.bst’ です．現在のバージョンは上記文書の 13th Edition に対応したものです．いずれのスタイルファイルも，オプションに指定するだけですから，ここでは例等は割愛します．

12.7 複数の索引を作りたい

異なる種類の索引を別々に作成して文末に置きたい場合があります．人名と語句の索引が別々になっている本は多く見受けられます．これを補助するものに ‘index.sty’[101] があります．この用途以外に，簡便な索引指定法も定義されています．使っていないので例は省略します．

種々の便利なマクロ

13.1 日付や曜日を出したい

これ関連のファイルはたくさんあるようですが，

まず ‘ukdate.sty’ というスタイルファイルがあります．これを用いると \today が英国式になります．曜日も \dayofweek で出力できます．つまり，今日は ‘Sunday $10^{\underline{\text{th}}}$ April 1994’ で，曜日は ‘Sunday’ と出力されます．序数関連で 1\st が ‘$1^{\underline{\text{st}}}$’，2\nd が ‘$2^{\underline{\text{nd}}}$’， 3\rd が ‘$3^{\underline{\text{rd}}}$’ さらに 5\th が ‘$5^{\underline{\text{th}}}$’ となる等の表示の他，\phaseofmoon が ‘new’ といった文字列を出力します． 漢字の曜日はこれを書き直すか，NTT jTEX に付属の ‘dow.tex’[102] 等を参考にするとよいでしょう．

他に曜日を出力するものなら ‘weekday.sty’ というのもあります． \weekday{2001}{01}{01} が Tuesday となり， \weekdaydate{2001}{01}{01} が Tuesday, January 1, 2001 と出力されます．

‘austdate.sty’ はオーストラリア式日付の表示をします．つまり今日は 10th April 1994 となります．

13.2 時刻を出力するマクロ

もともと \time にコンパイル開始時刻を記憶していますから，それを加工すればいいのです．そこで，

```
% \mod<counter><value> : <counter>=<counter> % <value>
\def\mod#1#2{
  \count255=#1 \divide\count255 by #2   % <register> = <counter> / <value>
  \multiply\count255 by #2              % <register> *= <value>
  \advance\count255 by -#1              % <register> -= <counter>
  #1=-\count255}                        % <counter> = -<register>
\newcount\hour
\hour=\time \divide\hour by 60          % <time> = <\time> / 60
\newcount\minute
\minute=\time \mod{\minute}{60}         % <minute> = <\time> % 60
```

[100] Kate L. Turabian による A Manual for Writers of Term Papers, Theses, and Dissertations かなと思ったのですが違うようです．この本も有名で出版社が The University of Chicago Press ですが，本物はもっと厚いでした．

[101] ‘index.sty’ はここにインストールされてないらしい．

[102] ‘dow.tex’ はここにインストールされてないらしい．

とすれば，文章中で \the\hour あるいは \the\minute で時刻を表示 (只今 21:11) できます．ただし，下に紹介するスタイルファイルによるものと異なり，このマクロは正六時などの場合には 6:0 と出力してしまいます．

日付や時刻を出力するものとしては他に ‘daytime.sty’ があり，ふたつのコマンド \daytime, \Daytime がそれぞれ 09:11pm, 9:11pm を出力します (10 時前にコンパイルしないと違いは現われません)．

さらに，‘time.sty’ もあります．こちらは \now が 9:11 P.M. になります．

また ‘mickeytime.sty’ も傑作です．次のようになります． “Now Mickey’s little hand is between the nine and the ten. Micky’s big hand is just past the two. Mickey’s eyes are open. ”

13.3　キーワードを脚注の位置に表示したい

例えば次のようなマクロでできそうです [5]．

```
\def\keyword#1#2{\def\@thefnmark{}\@footnotetext{{\bf #1}: #2}}
```

使用法は \keyword{キーワード}{かぎとなることば} のようにすれば，キーワードを脚注に載せることができます [‘latex.tex’]．

13.4　均等割り付けをしたい

これは文献 [7] の定義を変更した上で，さらに NTT jTEX および JaWaTEX でも使えるようにしたものです．

```
\def\kintou#1#2{%
 \setbox\@tempboxa\hbox{#1}\makebox[\wd\@tempboxa]{%
 \@ifundefined{jintercharskip}{%
  \kanjiskip=0pt plus 1fill minus 1fill
  \xkanjiskip=\kanjiskip #2}{%
  \jintercharskip=0pt plus 1fill minus 1fill
  \jasciikanjiskip=\jintercharskip #2}}}
```

使う場合には 2 つの引き数を必要としますが，最初のが割り付け幅を決める文字列， 2 つ目のが実際の文字列です．つまり「\kintou{この幅}{文字}」とすれば「文　字」となるわけです． 第 9.3 節 (p. 47) の例のようなコンパの案内等の箇条書き見出し等に便利かと思われます． 表 –**11** に均等でない場合と並べて比較してみました．割付巾は 6 文字です．

13.5　欧文での序数を自動的に出力したい

第 13.1 節 (p. 76) の英国日付のようなものを簡単に出力したい場合に便利なマクロがあります．これもあの Donald Arseneau 氏によるもので， comp.text.tex の記事です．マクロ部分を ‘ordinalno.sty’ というファイルに保存してあります．例えば \nth{1} や \nth{82} が 1st, 82nd 等となるマクロです．

13.6　数字の千の位にコンマを付ける

数字の千の位，あるいは 10 の 3 乗毎にコンマを付けることがあります．これを自動化すると，変更や修正が楽になります．ひとつは ‘sumofmoney.sty’ があります[103]．これを使えば \Number{1234567} が 1,234,567

キーワード: かぎとなることば．これは \keyword を使った例．

[103] 作者がつけたファイル名ではありません．

表–**11**　均等割り付けの例 (右 3 つは無意味です)

均 等 割 付	右寄せ	中央	左寄せ
造	造	造	造
土　　　木	土木	土木	土木
地　震　動	地震動	地震動	地震動
構 造 工 学	構造工学	構造工学	構造工学
粒 状 体 力 学	粒状体力学	粒状体力学	粒状体力学
コンクリート	コンクリート	コンクリート	コンクリート
安全性の評価法	安全性の評価法	安全性の評価法	安全性の評価法

になります．もうひとつは ‘numbersty.sty’ です．こちらは多機能で，上と同じ目的には \milnumber を使えば \milnumber1234567 が 1,234,567 になります．さらに \hexnumber1234 が 4D2 に変換されます．

13.7　音符や楽譜を書きたい

これには MusicTEX というものが存在[104]します．もともと plain TEX 用のマクロ群ですが，LATEX 用のスタイルファイルもあります．音符のメタフォントデータも含まれており，フォント自体もあちこちに置いてあると思います．使い方については付属のマニュアルを読んでください．このマニュアルは日本語版もあります．

13.8　袋綴じで出力したい

こんなことは，通常の出力をしておいてコピー機械で制御すればいいと思うのですが (両面綴じなどでは時折間違ってしまいますから)，頁数を考慮した上で文字縮小等を併用して自動的に希望の形式に出力するための支援スタイルファイルがあります．‘2up.sty’ (ここにあります) というものですが詳しい使い方はドキュメントを参照してください．プリンタ等の標準の環境で袋綴じができます．preamble に

```
\target{\magstep0}{2.1\textwidth}{\textheight}
\source{\magstep0}{\textwidth}{\textheight}
\targetlayout{twosided}
```

等とします．

13.9　OHP 用トランスパレンシーを作成したい

これも通常の出力をした上でコピー機械の拡大機能を使えばできますが，寸法の設定が意外と面倒なものです．これを紙の大きさと拡大率とを与えることによって，ある程度自動的にシートができるような支援スタイルファイルもあります．もちろん SLITEX を使うのが本来の仕様のようですが，数式等のフォントのインストール等のことを考えて，通常の LATEX で作成できた方が便利です．これには ‘eslides.sty’ (ここにあります) というファイルがあります．文字の拡大率は \magnification で定義し，出力するときにその倍率をドライバに指示します．表–**12** に倍率と dviout/prt Version 2.39 を用いた場合のスイッチとを示しました．

[104] akiu:ftp/pub/tex/misc/musictex/ の中．

表 –**12** ‘eslides.sty’ による OHP 作成時の設定

\magnification	1000	1095	1200	1440	1728	2074	2488
dviout/prt	-mag0	-maghalf	-mag1	-mag2	-mag3	-mag4	-mag5

\pagestyle を myslide にし，

```
\begin{slide}
 ...... for one tansparency
\end{slide}
\begin{remark}
  ...... comment and memo for that sheet
\end{remark}
```

のように slide 環境に入る部分が上下にセンタリングされた状態で仕上がります．頁番号は右下隅に出力されます．またすぐ続けての remark 環境にはメモも印刷できますが，この分はもちろん頁番号が付きません．さらに \logo というコマンドの再定義で左下隅に各人のロゴが入れられ，\conference の再定義により下中央に講演会名等を表示できます．

もっと細かく設定できるものに，‘seminar.sty’ (ここにあります) があります．非常に便利になっており，$\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX 等でも使える等の利点は多いのですが，マニュアルを読むのが面倒なので使っていません．

また ‘lslide.sty’[105] というのもありますが，これも使っていませんので割愛します．

13.10 ポスターを作りたい

ポスターのように大きな紙に大きな文字で出力したい場合があります．これには ‘poster.tex’ (ここにあります) というのが便利です．このマクロを使うと，最終的な出力を例えば A4 サイズの紙にクロップマーク付きで分割出力してくれます．印刷後に，そのクロップマークに添って切断・貼付すれば大ポスターが出来上がります． A4 で横 5 枚，縦 6 枚のポスターを作成した経験では，貼り合わせるのが若干たいへん (床に並べるので肉体的に) ですが出来上がりは結構いいものになりました．ドライバによっては印刷できないかもしれませんが，dviout/prt Version 2.39 なら -newsize オプションにする必要があります．使い方等はドキュメントと付属例を参考にしてください．

特に dviout/prt Version 2.41 で内臓フォントを用い第 14.3 節 (p. 82) にあるような大きな文字を用いる場合，プリンタの特性が原因かどうか分かりませんが，用紙をはみ出す文字が印字されない場合があります．このような場合には

```
-p=1;...;E
```

とオプションに E を追加してください．時間はかかりますが，隣り合う紙にまたがる文字もきちんと出力されます．

13.11 必要な頁だけをコンパイルしたい

dviware に頁毎の出力機能が無い場合にのみ必要だと思うのですが，ほんの一部を変更しただけの長い文書をコンパイルする場合には便利かもしれません．‘selectp.sty’ (ここにあります) を使います．出力したい頁を

---

[105] ‘lslide.sty’ はここにインストールされてないらしい．

```
\outputonly{1,3, 1 2-5}
```

のように指定します．この例では目次頁の 1, 3 頁目と，本文の 1 頁目および 2 から 5 頁が出力されます．この設定法についてはコメント欄を参照してください．

フォントの種類はどのくらいある

14.1　どんなフォントがあるか見たい

多分 ‘fntbl.tex’ か ‘testfont.tex’ がインストールされていると思いますから，例えば，

```
prompt > tex fntbl ⏎
```

とすれば，会話形式で見たいフォント表のファイル出力してくれます．あるいは，次の plain TeX の文書はフォントサンプルを表示します． LaTeX を使う場合は前後に定型の手続きが必要です．通常はありそうなフォントについては 10pt のものだけを 表 –**13** に示しておきました[106] (この ‘A quick ...’ にはすべてのアルファベットが含まれている)．

```
\def\fox{A quick brown fox jumps over the lazy dog.\par}
\def\sampleof#1{\font\fontA=#1 \rm #1:\quad\fontA \fox}
\sampleof{cmb10}        % Computer Modern Bold Roman
\sampleof{cmbsy10}      % Computer Modern Bold Math Symbols
\sampleof{cmbx10}       % Computer Modern Bold Extended Roman
\sampleof{cmbxsl10}     % Computer Modern Bold Extended Slanted Roman
\sampleof{cmbxti10}     % Computer Modern Bold Extended Text Italic
\sampleof{cmcsc10}      % Computer Modern Roman Caps and Small Caps
\sampleof{cmdunh10}     % Computer Modern Dunhill Roman
\sampleof{cmex10}       % Computer Modern Math Extension
\sampleof{cmff10}       % Computer Modern Funny Roman
\sampleof{cmfi10}       % Computer Modern Funny Italic
\sampleof{cmitt10}      % Computer Modern Italic Typewrite Text
\sampleof{cmmi10}       % Computer Modern Italic Math Italic
\sampleof{cmmib10}      % Computer Modern Math Italic Bold
\sampleof{cmr10}        % Computer Modern Roman
\sampleof{cmsa10}       %
\sampleof{cmsl10}       % Computer Modern Slanted Roman
\sampleof{cmsltt10}     % Computer Modern Slanted Typewriter Text
\sampleof{cmss10}       % Computer Modern Sans Serif
\sampleof{cmssbx10}     % Computer Modern Sans Serif Bold Extended
\sampleof{cmssdc10}     % Computer Modern Sans Serif Demibold Condensed
\sampleof{cmssi10}      % Computer Modern Slanted Sans Serif
\sampleof{cmsy10}       % Computer Modern Math Symbols
\sampleof{cmtcsc10}     % Computer Modern Typewrite Caps ans Small Caps
\sampleof{cmtex10}      % Computer Modern TeX extended ASCII characters
\sampleof{cmti10}       % Computer Modern Text Italic
\sampleof{cmtt10}       % Computer Modern Typewriter Text
\sampleof{cmu10}        % Computer Modern Unslanted Italic
\sampleof{cmvtt10}      % Computer Modern Variable-Width Typewriter Text
\bye
```

[106] もしかしたらとんでもないことをしているかもしれない．これ以降 \tt フォントが \fonts@mple フォントになっているような気がするが，どこがおかしいのだろう．

表–13 標準的な TeX のフォントサンプル

font	サンプル	その記述
cmr10	A quick brown fox jumps over the lazy dog.	Roman : `\rm`
cmti10	*A quick brown fox jumps over the lazy dog.*	Italic : `\it`
cmcsc10	A QUICK BROWN FOX JUMPS OVER THE LAZY DOG.	Small Caps : `\sc`
cmsl10	*A quick brown fox jumps over the lazy dog.*	Slanted : `\sl`
cmtt10	`A quick brown fox jumps over the lazy dog.`	Typewriter : `\tt`
cmbx10	**A quick brown fox jumps over the lazy dog.**	Boldface : `\bf`
cmss10	A quick brown fox jumps over the lazy dog.	Sans Serif : `\sf`
cmbxsl10	***A quick brown fox jumps over the lazy dog.***	Boldface Slanted
cmbxti10	***A quick brown fox jumps over the lazy dog.***	Boldface Italic
cmssbx10	**A quick brown fox jumps over the lazy dog.**	Boldface Sans Serif
cmssi10	*A quick brown fox jumps over the lazy dog.*	Italic Sans Serif
cmu10	A quick brown fox jumps over the lazy dog.	Unslanted Italic
cmssdc10	**A quick brown fox jumps over the lazy dog.**	S S Demibold Condensed
cmitt10	*A quick brown fox jumps over the lazy dog.*	Italic Typewriter
cmsltt10	*A quick brown fox jumps over the lazy dog.*	Slanted Typewriter
cmtcsc10	A QUICK BROWN FOX JUMPS OVER THE LAZY DOG.	Typewriter Small Caps
cmvtt10	A quick brown fox jumps over the lazy dog.	Variable-Width Typewriter
cmtex10	A quick brown fox jumps over the lazy dog.	TeX Extended `ASCII` Char
cmdunh10	A quick brown fox jumps over the lazy dog.	Dunhill
cminch	ABC	1-Inch-High Char
cmfi10	*A quick brown fox jumps over the lazy dog.*	Funny Italic
cmff10	A quick brown fox jumps over the lazy dog.	Funny Roman
cmmi10	$Aquickbrownfoxjumpsoverthelazydog$	Math Italic
cmmib10	$\boldsymbol{Aquickbrownfoxjumpsoverthelazydog}$	Math Italic Boldface
cmsy10	$\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$	Math Symbol
cmbsy10	$\boldsymbol{\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}}$	Boldface Math Symbol

14.2 標準ではロードされていないフォントを使ってみたい

通常プレロードされていないフォントを使いたいことがあります．これには `\newfont` コマンドで呼び出せばいいだけです [5]．例えば boldface italic 体 (`cmbxti10`) を 12pt で使いたい場合は

```
\newfont{\bolditalic}{cmbxti10 scaled \magstep1}
```

とします．しかし，これでは種々の大きさを \Large 等のコマンドでは自由に変更できません．これをする方法は文献 [8] にありますが，スタイルファイルもあります．オリジナルは ‘bsf.sty’ (ここにあります) というスタイルファイルで，これは boldface の Sans Serif 体をひとつのフォントファミリーに登録するものです．これを利用すれば，

```
\documentstyle[bsf]{article}
...
This is normal. {\Large\bsf Bold Sans Serif font} Again normal
```

というように，普通の字体と同じ様に使えます． これを真似して boldface slanted, boldface italic, Dunhill, slanted Sans Serif, funny italic 体の定義もしてみました． ‘fontfam.lzh’[107] にまとめてありますが，同時にいくつもは定義できません． ただ，いくつかはすでに存在したようで， ‘bsl.sty’[108] は boldface slanted のフォントファミリーを， ‘isf.sty’[109] は italic Sans Serif のファミリーを定義します． また ‘slem.sty’[110] は \em を \sl に置き換えます．

14.3 日本語のフォントを変更したい

明示的にフォントを指定するためには，右の表のようなコマンドを使いますが，通常はデフォールトが明朝体で， \bf でゴチック体になるようになっています．また，普通の文書ならその方が互換性があっていいでしょう．

	明朝体	ゴチック体
アスキー 日本語 TEX	\mc	\gt
NTT JTEX	\dm	\dg

また，「書体倶楽部」のようなベクトルフォントや，プリンタ内蔵のスケーラブルフォントが使える場合には，かなり大きな漢字も使えます．例えば NTT JTEX なら \jfont を使って

```
\jfont\fontA=dm10 scaled 9500
```

等としますと， {\fontA 漢字} で約 3cm 角の漢字になります．ゴチックの場合は dg10 ですが， JaWaTEX の場合にはそれぞれ m10, b10 になります．互換性が無いので原則として使わないようにしたいものです．しかし，第 13.10 節 (p. 79) のポスター等には便利です．

14.4 外字を作って使いたい

正当な方法はメタフォントで作成することですが，数文字だけならドットパターンを定義することによる外字を定義できるようです．スタイルファイル名は ‘sprite.sty’ というものです． 図 –**13** に示したように，好みのドット数で縦横のパターンを

```
\sprite{\macroname}(n,m)[width,height] ... \endsprite
```

で定義します． n, m がデータの縦横ドット数．それを， width, height の箱として定義します．同じドットパターンでも width, height の値によって大きさを制御できます．

[107] ‘fontfam.lzh’ はここにインストールされてないらしい．

[108] ‘bsl.sty’ はここにインストールされてないらしい．

[109] ‘isf.sty’ はここにインストールされてないらしい．

[110] ‘slem.sty’ はここにインストールされてないらしい．

```
\newsavebox{\GAIJIa} \sbox{\GAIJIa}{>8}
\def\SNIPSNIP{\kern.5pt%
 \raisebox{-.18ex}{\scissors}\kern 1pt}
\sprite{\scissors}(24,24)%
          [1.2\wd\GAIJIa,1.5\ht\GAIJIa]
     :BB...................... |
     :.BB..................... |
     :..BBB.............BBBB.. |
     :...BBBB..........BB..BB. |
     :....BBBBB.......BB....BB |
     :.....BBBBB.....BB.....BB |
     :......BBBBB...BB......BB |
     :.......BBBBB..BB......B. |
     :........BBBB.BB......BB. |
     :.........BBBBBB.....BB.. |
     :..........BBBBB....BB... |
     :...........BBBBBBBBB.... |
     :...........BBBBBBBB..... |
     :..........BBBBBBBBBBB... |
     :.........BBB.BB.....BB.. |
     :........BBBB.BB......BB. |
     :.......BBBB..BB.......B. |
     :......BBBBB...BB......BB |
     :.....BBBBB....BB......BB |
     :....BBBBB......BB.....BB |
     :...BBBBB.......BB....BB. |
     :..BBBB..........BB..BB.. |
     :.BBB.............BBBB... |
     :BB...................... |
\endsprite
```

```
\sbox{\GAIJIa}{()}
\def\DYNAMITE{\kern.5pt%
 \raisebox{-.18ex}{\Dynamite}\kern 1pt}
\sprite{\Dynamite}(24,24)%
          [1.5\wd\GAIJIa,1.5\ht\GAIJIa]
     :........................ |
     :.................B...... |
     :........................ |
     :...........B....B...B... |
     :.............B.....B.... |
     :.......................B |
     :...............B....B... |
     :..............BBB....... |
     :.....BBBBBBB.BBB...B.... |
     :...BBBBBBBBBBBBB.......B |
     :..BBBBBBBBBBBBB......... |
     :.BBBBBBBBBBBBBBB........ |
     :.BBBBBBBBBBBBBBB........ |
     :.BBBBBBBBBBBBBBB........ |
     :BBBBBBBBBBBBBBBBB....... |
     :BBBBBBBBBBBBBBBBB....... |
     :BBBBBBBBBBBBBBBBB....... |
     :BB..BBBBBBBBBBBBB....... |
     :.BBBBBBBBBBBBBBB........ |
     :.BB..BBBBBBBBBBB........ |
     :..BB..BBBBBBBBB......... |
     :...BB..BBBBBBB.......... |
     :....BBBBBBBBB........... |
     :.......BBB.............. |
\endsprite
```

例えば「熈」のような人名漢字や，『危険！ \DYNAMITE』が『危険！ 』となる等の遊びも，メタフォントを利用せずに可能です．

```
\noindent\mbox{}\dotfill\dotfill\raisebox{-3pt}[0pt][0pt]{\SNIPSNIP}%
 \dotfill\raisebox{-1.5pt}[0pt][0pt]{\ {\small cut here}\ }\dotfill\dotfill\mbox{}
```

図 –**13** sprite.sty で外字を定義して使う例

.... 切り取り線 ....

14.5　標準ではないフォントを使ってみたい

14.5.1　左右矢印の代わりに手の平のフォントはありませんか

ありました．主なサイズの pk フォントも含めてありますが， handsfnt というものです． 例えば，

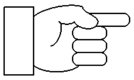
```
\font\font@hands=hands scaled \magstep0 % for example
\font\font@bhands=handsbig
\def\handsright{\ \raisebox{.5ex}[0pt]{\font@hands\symbol{"41}}\ }
\def\bighandsright{\ {\font@bhands\symbol{"41}}\ }
\def\handsleft{\ \raisebox{.5ex}[0pt]{\font@hands\symbol{"42}}\ }
\def\bighandsleft{\ {\font@bhands\symbol{"42}}\ }
\def\handsrightr{\ \raisebox{.5ex}[0pt]{\font@hands\symbol{"43}}\ }
\def\bighandsrightr{\ {\font@bhands\symbol{"43}}\ }
\def\handsleftr{\ \raisebox{.5ex}[0pt]{\font@hands\symbol{"44}}\ }
```

表 –**14** カリグラフィック文字や古いドイツ文字の例

font	サンプル	その記述
`manfnt`	METAFONT METAFONT	manual font
`callig15`	*A quick brown fox jumps over the lazy dog.*	Calligraphy
`yfrak`	Digitales Tonbandgerät bei der Aufnahme, bei ...	Fraktur
`ysmfrak`	Digitales Tonbandgerät bei der Aufnahme, bei ...	Fraktur
`ygoth`	Digitales Tonbandgerät bei der Aufnahme, bei ...	Gothisch
`yswab`	Digitales Tonbandgerät bei der Aufnahme, bei ...	Schwabacher
`yinit`	ABCDE	Initials

```
\def\bighandsleftr{\ {\font@bhands\symbol{"44}}\ }
```

のようなマクロを定義しておけば，

handsbig	symbol	hands	symbol	hands	symbol
`\bighandsright`		`\handsright`	☞	`\handsleft`	☜
		`\handsrightr`	☞	`\handsleftr`	☜

この表のような使用が可能となります．ＯＨＰでの強調等に便利かもしれません．

14.5.2 TEX のマニュアルフォントはありませんか

TEXBook[9] で使われているフォントですが，いわゆる `manfnt` というフォントがそれです． `logo` フォントと同様に `METAFONT` のロゴや危険サインがあります．例は表 –**14**に示してありますが，他にもいろいろあります．

14.5.3 カリグラフィック文字はありませんか

和名は「書道体」とでも呼ぶのかもしれませんが，いわゆるカリグラフィと呼んでいる文字で，昔の羽根ペンで書く筆記体のような文字です．例は表 –**14**に示してあります．文書全体をこの文字で出力する時の便宜を計るために，第 14.2 節 (p. 81) のスタイルファイルと同様に，種々の大きさを通常のコマンドで指定できるようにもしてみたのが ‘`calligra.sty`’ (ここにあります) です．

14.5.4 古いドイツ文字はありませんか

あります． ‘`old-german`’ というのが総称です．国内関東側 `archie` でみつかるサイトのは，一部のファイルが壊れていたりバグがあったりするようですので，適切に修正する必要があります．フォント種は表 –**14**に

示した 5 種類ですが，それぞれ 'Fraktur', 'Gotisch', 'Schwabacher' および 'Initials' です．ウムラウト等の入力にそれなりの規則がありますから付属のドキュメントを読んでください．

14.6 標準のシンボル表はありませんか

LaTeX マニュアルや各種本にある，通常文中での記号と数式モードでの記号を表にしたものがあると便利です． AMS-TEX のフォントも合わせて 'symbols.tex'[111] というファイルをコンパイルすれば，現状で使える記号表が出来上がります． 同様のものに 'latexsymbols.tex'[112] というものもあります．一応，第一筆者が雑用文書に必要なものだけを集めた表に，不足分を書き加えたものを巻末に置きました．順番はマニュアル等とは異なっています．

14.7 標準のフォントで記号等を作る

14.7.1 各種記号

特殊文字や時折必要と思われるものを文献等から引用しておきます．もちろん AMS-TEX の利用やメタフォントによるフォント作成という手段はありますが，ここでは標準の LaTeX の範囲内での利用を考えています．

- $\therefore$ 渡辺氏作 $\left\{\begin{array}{l}\texttt{\textbackslash def\textbackslash therefore\{\textbackslash setbox0 \textbackslash hbox\{\$\textbackslash cdot\$\}} \\ \texttt{\textbackslash raise-0.2em \textbackslash copy0 \textbackslash raise0.2em \textbackslash copy0 \textbackslash raise-0.2em \textbackslash box0 \~{}\}}\end{array}\right.$
- $\because$ 渡辺氏作 $\left\{\begin{array}{l}\texttt{\textbackslash def\textbackslash because\{\textbackslash setbox0\textbackslash hbox\{\$\textbackslash cdot\$\}} \\ \texttt{\textbackslash raise0.2em \textbackslash copy0 \textbackslash raise-0.2em \textbackslash copy0 \textbackslash raise0.2em \textbackslash box0 \~{}\}}\end{array}\right.$
- $\boxed{\hookleftarrow}$ `$\Box$` と `$\hookleftarrow$` との重ね合わせで作りましたが， 'ascmac.sty' や文献 [7] のマクロの方が格好がいいでしょう．

また " で “ や ” を出力するものに 'quote.sty' があります．常にペアで存在しなければならない制限はありますが， `\begindoublequotes` から `\enddoublequotes` までで有効になります．他のソフトで作成した文書を読み込む場合等に便利かもしれません．例えば `"quotation mark"` が “quotation mark” になります．

14.7.2 丸に入れた数字

これは標準文字列表には定義されていませんので，文献 [8] の方法で合成して使うのが一番簡単です．それを参考にしながら，渡辺さんの重ね書きのマクロをそのまま使ったのが，

```
\def\MARU#1{\leavevmode\setbox0\hbox{$\bigcirc$}%
            \copy0\kern-\wd0 \hbox to\wd0{\hfil{\scriptsize#1}\hfil}}
```

というものです．例えば `\MARU{3}` 等が ③, ㊹ になります．

14.7.3 ℃ や ¥ はどうするか

まず ¥ マークですが，これは次のような方法があります．

- ¥ `\def\yen{{\leavevmode\tt\rlap=Y}}`
- ¥ `\newlength{\Ywidth}`
  `\def\yen{\mbox{\tt\settowidth{\Ywidth}{Y}Y\hspace{-\Ywidth}=}}`

[111] 'symbols.tex' はここにインストールされてないらしい．

[112] 'latexsymbols.tex' はここにインストールされてないらしい．

一方， ˚C は次のような方法があります．

˚ C　`\newcommand{\DegC}{\char'27 C}`

˚C　`\newcommand{\DegC}{\char'27\kern-.3em\hbox{C}}`

上付き丸と C との距離は各人の好みで設定してください．

14.7.4　スマイリー

これも TEX and TUG NEWS (Vol.1, No.4 Nov 1992 ‘ttn1n4.tex’) にあったもので， Peter Flynn 氏 (University College Cork, cbts8001@iruccvax.ucc.ie) によるものです．ただし，オリジナルをちょっと変更して，何度でも使えるようにしました．つまり，

```
\typeout{Smiley by Peter Flynn (cbts8001@iruccvax.ucc.ie)}
\font\tmi@grin=cmmi7
\font\tsy@grin=cmsy5
\font\bsy@grin=cmsy10 scaled\magstep4
\newsavebox{\sbox@grin}
\savebox{\sbox@grin}{\vbox{\hsize=1pc\baselineskip=5pt
                    \centerline{{\tsy@grin\char'014\ \char'014}}
                    \centerline{{\tsy@grin\char'064}}\vskip-1pt
                    \centerline{{\tmi@grin\char'136}}}}
\newsavebox{\bbox@grin}
\savebox{\bbox@grin}{\hbox{{\bsy@grin\char'015}}}
\def\grin{\hbox{\usebox\bbox@grin\kern-5.75mm\usebox\sbox@grin%
          \kern-\wd\sbox@grin\kern5.75mm}}
```

としてみました．単に \grin とすれば となります．水平の位置合わせがいい加減です．使うときは引用について記しておいてください．

14.8　各種ロゴマーク

もともとの LATEX のロゴは \sc を使った \rm で定義されているので，太字やイタリックにはなりません．ですから，特に節タイトル等に含めた場合に見劣りがします．これを改善する方法は Network News でも種々提案があるようですが，ひとつに ‘ttn2n1.sty’[113]にあった定義が便利そうでしたので，この文書でも使ってみました．

それ以外のロゴマークについてもいくつか定義してあります． あるいは ‘texnames.sty’[114] にも定義がありますし， 確か NTT jTEX に付属の ‘texsym.sty’ (ここにあります) も同じ目的のファイルです．

その他， TEX, LATEX プログラミングにも関連して

15.1　NTT jTEX と アスキー 日本語 TEX，何と厄介な！

2 つの規則があることはユーザーにとっては厄介なことです．しかも，ファイル先頭の書き出しから異なるのですから困ったものです．この文書はそのどちらでも (多分) できるようにはしてみましたが，ここで用いた方法は ‘j-article.sty’ を探して，もしあったら NTT jTEX と判断するもので，

[113] ‘ttn2n1.sty’ はここにインストールされてないらしい．

[114] ‘texnames.sty’ はここにインストールされてないらしい．

```
\makeatletter \newread\@istheresuchafile
\openin\@istheresuchafile j-article.sty
\ifeof\@istheresuchafile \relax
  \documentstyle[styleuse]{jarticle}         % ASCII
\else\closein\@istheresuchafile \relax
  \documentstyle[styleuse]{j-article}        % NTT
\fi \makeatother
```

としてみました．もちろん，もっといい方法はあるようです．ひとつ挙げると，

```
\ifx\gtfam\undefined
  \documentstyle[styleuse]{j-article}        % NTT
\else
  \documentstyle[styleuse]{jarticle}         % ASCII
\fi
```

です．ゴチック漢字のフォントファミリー定義の存在で判断してあります．マクロも両用できるようにする時には，このような区別が必要になります．この文書 (第 13.4 節 (p. 77)) で文献 [7] の定義を拡張した均等割り付けも漢字同士のスペーシング `\jintercharskip` の定義の有無で判断しました．これは アスキー 日本語 TEX では `\kanjiskip` に相当します．

15.2 キーボードから入力したい

コンパイルのある部分を会話形式で制御したい場合があります．その場合には `\typein` コマンド [5] を使います．例えばコンパイルエラーだけの検出が目的なのか，最終的な dvi ファイルの出力が必要なのかを会話形式で決定するようにするには，

```
\typein[\answer]{Do you need dvi output? (y or n) }
\if\answer y
 \typeout{Normal operation with output}
\else
 \typeout{OK! No output} \output={\setbox0=\box255\deadcycles=0}
\fi
```

で制御できます．ここの選択は Yes, No の 2 つだけですが，ここに複数選択肢メニューを表示して選択させるようにすることもできます．この方法については第 3.9 節 (p. 15) の ‘fillform.tex’ (ここにあります) の中を参照してください．

15.3 オプション付きのコマンドを定義したい

ブラケット “[...]” で囲まれたオプション等を許すマクロを書くには `\@ifnextchar` を用います．この場合は，次のような構文になります．

```
\def\foo{\@ifnextchar [{\@foo}{\@@foo}}
 \def\@foo[#1]#2{.....}   % with option [...]
 \def\@@foo#1{.....}      % w/o  option
```

あるいはこの [..] に入るものにデフォールトがあるような場合，例えば 2em という長さがデフォールトである場合，

```
\def\foo{\@ifnextchar [{\@foo}{\@foo[2em]}}
 \def\@foo[#1]#2{.....}           % ^^^^^ = default
```

とできます．\@ifnextchar が { の場合にはこの { をマクロには直接書けませんから，\bgroup という文字列を使います [8]．

また * がコマンドに付くか否かで機能を分けたいような場合には，\@ifstar{yes}{no} を使います．

```
\def\foo{\@ifstar{\@foostar}{\@foonostar}}
```

等としておけば，\foo{...} と \foo*{...} とに別機能を与えられます．

15.4　if による選択を随所でしたい

例えば英文の場合と和文の場合で書式が異なる場合，アスキー日本語 TEX と NTT jTEX とで異なる場合，あるいは草稿段階と清書とで区別をしたい場合等いろいろ考えられます．この場合には新しい判断制御を宣言すればいいのです．つまり，例えば

```
\newif\ifdraft
```

と宣言しておくと，この draft を true にしたり false にしたりすることによって制御の流れを変える判断に使えるようになります．例えば

```
\documentstyle{article}
\newif\ifdraft          % newly define \ifdraft
\drafttrue              % Now draft mode ON!!!
\begin{document}
....
\ifdraft
        ...             % operation when draft mode is ON
\else
        ...             % operation at the final manuscript
\fi
...
```

といった具合です．第 15.1 節 (p. 86) の判断と一緒に使えば，アスキー日本語 TEX, NTT jTEX での使い分けが簡単にできそうです．

15.5　\@tfor の変形

コンマで区切ったリストを順に出力する \@tfor は単純にそのリストを出力するだけです．土木学会論文集のスタイルファイルを作成している際に，英文で作成した場合に例えば \fignos{2,5,6,9} が **Figs. 2, 5, 6** and **9** にならないかという相談を渡辺さんにして作ってもらったのが次のマクロです．つまり，列挙したリストの最後を判別して処理して欲しかったのです．実は第一筆者は全くその中身を理解できないのですが，

```
\def\andfor#1{\@ffor\@ndfora:=#1\do{\@ndfora}}
\def\@ffor#1:=#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\@empty \else
  \expandafter\@fforloop#2,\@nil\@@#1{#3}\fi}
\def\@fforloop#1,#2\@@#3#4{\def#3{#1}\ifx #3\@nnil \else
  \ifx #2\@nil{#4}\else
  {#4}\@ifforloop #2\@@#3{#4}\fi\fi}
\def\@ifforloop#1,#2\@@#3#4{\def#3{#1}\ifx #2\@nil
  {\ and\ {\ignorespaces #4}}\let\@nextwhile=\@fornoop \else
  {,\ {\ignorespaces #4}}\let\@nextwhile=\@ifforloop\fi
  \@nextwhile#2\@@#3{#4}}
```

としておけば，例えば \andfor{apple, banana, orange} という表現は apple, banana and orange になります．うぅーん，複雑過ぎてよく分からない．この方法は第 4.5 節 (p. 19) でも紹介した土木学会論文関連スタイルファイルと第 3.10 節 (p. 15) の 'cllctdpp.sty'[115]の著者名表示にも使っています．

15.6 Overfull/Underfull hbox 警告を減らしたい

特に NTT JTeX の場合は，漢字の文字間隔が厳密すぎるために Overfull, Underfull の警告が出やすいようです．次のように，その値を少しルーズにすると警告が少なくなることがあります．

```
\jintercharskip 0.5mm plus 0.1mm minus 0.3mm
```

これで文字間隔が，目標としてはおよそ 0.2mm ~ 0.6mm になります (上限はない [8]) が，あまり範囲を拡げ過ぎると行によっては文字間隔が違い過ぎてしまいます．頻繁に出る可能性がある行は英数字が含まれている行のようですから，英数字と漢字との間にこまめに半角スペースを入れるのも手かもしれません．少しくらい警告が出ても平気な人もいますから，あとは個人の好みでしょうか．

これでおしまい

16.1 自分のサイトのスタイルファイルのありか

LaTeX は環境変数 TEXINPUTS (もしかすると最後の S は無い場合もあります (emTeX)) に設定されたディレクトリに文書そのものやスタイルファイルを探しに行きます．自分専用のスタイルファイルを置くディレクトリを，例えば HOME/tex/inputs にしたいときは，

```
setenv TEXINPUTS ``.:$HOME/tex/inputs:/usr/local/lib/tex/inputs''
```

のようにします．この設定だと，まずカレントディレクトリ，続いて自分の $HOME/tex/inputs と続き，最後に /usr/local/lib/tex/inputs にスタイルファイルを探しに行くようになります．環境変数 TEXINPUTS が設定されていないときのデフォルトの path は

```
.:/usr/local/lib/tex/inputs
```

となっている場合が多いでしょう．もちろんサイトによって大きく異なりますから，詳細は管理者やその筋の人に尋ねてください．基本となるスタイルファイルや free で手に入りこの文書で紹介したようなものはほとんど同じディレクトリにあると思われます．

ですから，既存のマクロを変更したり追加したりしたい場合は，このディレクトリを捜しだし，目的のファイルを自分の作業領域にコピーすることから始めます．逆に言うと，勝手にファイルを書き変えたり，自分だけのスタイルファイルを共通の所に置くのは間違いの元になりますので避けてください．

16.2 スタイルファイルの使用サンプル

この文書でもいくつか挙げてみましたが，たいていのファイルには別ファイルあるいは \endinput の後に例題が付いています．それ以外のほとんどはコメント欄をよく読まなければなりません．また，'docstrip.tex' (ここにあります) という Mainz file のうちの MZ0 パッケージに含まれているユーティリティで，オリジナルの foo.doc ファイルからスタイルファイルとマニュアルおよび例題を取り出すものもあります．

[115] 'cllctdpp.sty' はここにインストールされてないらしい．

16.3 スタイルファイルをハックしたい

該当するディレクトリにある各種ファイルに，既存のマクロは定義されています．詳細は文献 [8] に譲ります．またハックする場合も，できれば文献等を読んだ上で行った方が効率がいいはずです．表–**15** に簡単に説明しました．現在 ‘plain.tex’ や ‘latex.tex’ の最新版は 25 March, 1992 の日付を有する TEX Version. 3.141 対応と思われます． LaTeX は Version 3 以前のままなのでその整合性は内部で処理してあるようですが， plain TEX のフォーマットファイル生成ではエラーが生じます[116]．

表–**15** 種々のファイルの簡単な解説

‘lplain.tex’	:	LaTeX の基本的なコマンド群
‘jlplain.tex’	:	日本語 LaTeX の基本的コマンド群
‘latex.tex’	:	LaTeX の最も一般的なコマンド群
‘article.sty’	:	欧文の article スタイル
‘jarticle.sty’	:	アスキー 日本語 TEX の日本語 article スタイル，あるいは NTT jTEX の旧版
‘j-article.sty’	:	NTT jTEX の日本語 article スタイル
‘art10.sty’	:	article スタイルを 10 ポイント用で用いるためのもの． ‘report.sty’, ‘book.sty’ なども同様

TEX の仕組みに興味があれば文献 [3] が， LaTeX のとっかかりには文献 [4] あたりがいいと思います．マニュアルは文献 [5] ですが，細かい設定等のデフォールト一覧等については文献 [6] がいいでしょう．文献 [7] は アスキー 日本語 TEX 関連のスタイルファイルの変更の仕方等が中心ですが， NTT jTEX ユーザーにとっても分かりやすいと思います．最終的にマクロの意味を正確に理解するには文献 [8] がいいようです． TEX の本当のマニュアルを読むのは，他力本願寺の宗徒にとっては苦痛だと思われます．

16.4 おわりに

他力本願寺宗徒のままでいるのが LaTeX 使いの本来の姿だと思いますし，その方が楽です．そのためにここでは多くのスタイルファイルを紹介してみました．かなり便利なものが既に揃っていますので，そのままあるいは少し手を加えるだけで望みの機能が得られます．ですからレイアウトのことは考えることなく，文書作成に専念できるわけです．ただ大抵は文章を書くこと自体がたいへんな苦痛で，ついつい自己逃避をしたくなるのも事実です．もしそのとき何かいいアイデアが浮かんだら，国民の皆の役に立つスタイルファイルを作って公開してください．

ただ，できる限り `\endinput` のあとに英文 (あるいは意味の無い半角文字の羅列で) のサンプルを付け (漢字によるコマンド名は特殊な場合を除いて避けた方がいいと第一筆者は考えています)，さらに アスキー 日本語 TEX, NTT jTEX の区別なく使えるようにコーディングして欲しいものです．

この春 LaTeX2$_\epsilon$ が公開されます (β 版はすでに公開されています) が，どういったものかはまだ把握しておりません．ここに示したマクロやスタイルファイルのいくつかは新しいシステムでは必要が無くなるものもありそうです．また，多くはそのままでは新しいシステムで使えないようです．ただ，この新しいシステムは，旧来の LaTeX 2.09 をサポートしていますので，この文書もしばらくは使えるとは考えています．

[116] 放っておいても構わないようだが．

表–16 ギリシャ文字 (数式モード)

α	\alpha	β	\beta	γ	\gamma	δ	\delta	ϵ	\epsilon	ζ	\zeta
η	\eta	θ	\theta	ι	\iota	κ	\kappa	λ	\lambda	μ	\mu
ν	\nu	ξ	\xi	π	\pi	ρ	\rho	σ	\sigma	τ	\tau
υ	\upsilon	ϕ	\phi	χ	\chi	ψ	\psi	ω	\omega		
ε	\varepsilon	ϑ	\vartheta	ϖ	\varpi	ϱ	\varrho	ς	\varsigma	φ	\varphi
Γ	\Gamma	Δ	\Delta	Θ	\Theta	Λ	\Lambda	Ξ	\Xi	Π	\Pi
Σ	\Sigma	Υ	\Upsilon	Φ	\Phi	Ψ	\Psi	Ω	\Omega		

表–17 数式モードで使える記号

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

参考文献

[1] Leslie Lamport, LaTeX —*A Document Preparation System*, Addison-Wesley Publishing Co., 1986.

[2] 小田部荘司，縄田毅史， LaTeX のスタイルファイルを変更する — ファミリーセダンをチューンアップ —，九大大計センター広報， 1989.

[3] 大野義夫， TeX 入門，共立出版， 1989.

[4] 野寺隆志，楽々 LaTeX，共立出版， 1990.

[5] Edgar Cooke，倉沢良一監訳， LaTeX[1], 大野俊治，小暮博道，藤浦はる美　訳，アスキー出版局，1990.

[6] 伊藤和人， LaTeX トータルガイド，秀和システムトレーディング， 1991.

[7] 奥村晴彦， LaTeX 美文書作成入門 — パソコンによる文書レイアウト —，技術評論社， 1991.

[8] 磯崎秀樹， LaTeX 自由自在， Computer Today ライブラリ — 5，サイエンス社， 1992.

[9] Donald E. Knuth, The TeXbook, Addison-Wesley Publishing Co., 1984.

[10] 斎藤信男監修， TeX ブック [9]，鷺谷好輝　訳，アスキー出版局， 1989.

表–18　二項演算記号 (数式モード)

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

表–19　関係記号 (数式モード)

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

表–20　矢印記号 (数式モード)

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

表 –21　大きさ可変の記号 (数式モード)

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

表 –22　関数等の記号 (数式モード)

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

表 –23　\left と \right とでバランスする記号・括弧 (数式モード)

$($	(	$)$	)	$[$	[	$]$	]
$\{$	\{	$\}$	\}	$\lfloor$	\lfloor	$\rfloor$	\rfloor
$\lceil$	\lceil	$\rceil$	\rceil	$\langle$	\langle	$\rangle$	\rangle
$/$	/	$\backslash$	\backslash	$\vert$			$\Vert$	|
$\uparrow$	\uparrow	$\downarrow$	\downarrow	$\Uparrow$	\Uparrow	$\Downarrow$	\Downarrow
$\updownarrow$	\updownarrow	$\Updownarrow$	\Updownarrow			.	(to balance)

表 –24　\[B]bi[g]gr[l] 等と使う大きな区切り記号 (数式モード)

$\rmoustache$	\rmoustache	$\lmoustache$	\lmoustache	$\rgroup$	\rgroup	$\lgroup$	\lgroup
$\arrowvert$	\arrowvert	$\Arrowvert$	\Arrowvert	$\bracevert$	\bracevert		

表 –25 通常モードおよび数式モードでのアクセント

出力	通常モード	数学モード
ò	\`{o}[117]	\grave{o}
ó	\'{o}[118]	\acute{o}
ô	\^{o}	\hat{o}
$\widehat{M}$		\widehat{M}
ö	\``{o}	\ddot{o}
õ	\~{o}	\tilde{o}
$\widetilde{M}$		\widetilde{M}
ō	\={o}[119]	\bar{o}
$\overline{o}$		\overline{o}

出力	通常モード	数式モード
ȯ	\.{o}	\dot{o}
ŏ	\u{o}	\breve{o}
ǒ	\v{o}	\check{o}
ő	\H{o}	
o͡o	\t{oo}	
o̧	\c{o}	
ọ	\d{o}	
$\vec{o}$		\vec{o}
o̲	\underline{o}	\underline{o}

[117,118,119]: tabbing 環境中ではそれぞれ \a`{o}, \a'{o}, \a={o} を使います .

表 –26 通常モードで使える記号

†	\dag	‡	\ddag	§	\S	¶	\P
©	\copyright	£	\pounds	ł	\l	Ł	\L
ß	\ss	¿	?`	¡	!`	#	\#
$	\$	%	\%	&	\&	_	_
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA	ø	\o	Ø	\O
{	\{	}	\}	ˆ	\tt\symbol{'136}		
\	\tt\symbol{'134}			˜	\tt\symbol{'176}		

さくいん

1. ¶マークが付いているファイルはこの文書中にサンプル (のようなもの) が付いてます．
2. †マークはオリジナルにサンプルファイルがあるもの．
3. ‡マークは docstrip.tex を使うもの．
4. §マークは \endinput のうしろにサンプルがあるもの．
5. 無印はコメント行に解説があるものになっています．