目 次

第 II 部　中級コース **1**

1　マクロの定義 **1**
1.1　単純な置き換え：引数なし 1
1.2　引数がある場合 3

2　フォントあれこれ **4**
2.1　非標準フォントのロード 4
2.1.1　日本語フォント 4
2.2　フォントの一覧 5
2.2.1　フォントを使った枠組み 5

3　様々なスタイルファイルの利用 **7**

4　レイアウトの情報 **11**
4.1　長さの値表示：\the 11

5　リスト処理 **12**

6　データファイルからの直接の作表 **13**

第 II 部
中級コース

中級コースでは、数式記号やスタイルファイルの自由な使いこなしと簡単なマクロの自作を練習します。

1　マクロの定義

ある命令の並びに名前をつけて、その名前で命令列を呼び出す機能をプログラミング言語では一般に、マクロといいます。一方、最初から言語に組み込まれている命令をプリミティブといいます。TEX では主に \def というプリミティブが使われますが、その命令が既に定義されているかチェックしません。従って、うっかり他で定義済の命令を再定義してしまう可能性があります。LATEX では、\def の他に\newcommand が使え、定義済かどうかチェックしてくれます。

1.1　単純な置き換え：引数なし

もっとも簡単な引数なしの置き換えマクロを練習しましょう。形式は極めて簡単です。

置き換えマクロ：引数なし

```
\def\マクロ名{内容}
```

例えば、

```
\def\TSF{Thin Solid Films}
\def\じゅげむ{じゅげむじゅげむごこうのすーりきっれ...}
```

と定義しておいて、文章中で

```
\TSF{}は\じゅげむ{}です。
```

と書けば

Thin Solid Films はじゅげむじゅげむごこうのすーりきっれ... です。

のように内容が展開されます。

練習 1　右欄の表示を得るように、左欄のマクロ定義を修正・完成させなさい。

マクロ定義	表示結果
\def\a-Si{amorphous Silicon	amorphous Silicon
\def\メタン{$ $}	CH_4
\def\自宅{\begin{tabular}[t]{@{}c@{}} 杉並区下高井戸 5-11-12　\bf 電話 03-3302-3488	杉並区下高井戸 5-11-12 電話 *03-3302-3488*
\def\ShrodFree1{-\frac{\hbar^2}{2m}　}	$-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\psi(x) = E\psi(x)$

最後のマクロは数式モードで使うことを前提に考えます。したがって、$...$ で囲みません。

LaTeX の実体は TeX のプリミティブを使ったマクロの集まりです。その他のスタイルファイルも同様です。そこでは、この単純な置き換えマクロも随分と定義されています。実際に ascmac.sty (/usr/local/lib.tex/macros にあります。) の中の円記号 ¥ のマクロ定義を解読してみましょう。

```
%% ¥
\def\yen{{\setbox0=\hbox{Y}Y\kern-.97\wd0\vbox{\hrule height.1ex width.98\wd0
\kern.33ex\hrule height.1ex width.98\wd0\kern.45ex}}}
```

ちょっと解り難いですから、コマンド毎に区切ります。

```
%% ¥
1. \def\yen{          : マクロ \yen の定義開始宣言
2. {\setbox0=\hbox{Y} : 文字 Y の大きさ（箱）を box0 に収める
3. Y                  : 文字 Y（の表示）
4. \kern-.97\wd0      : wd0(=box0 の幅) の-0.97 倍の長さの空白を入れる
5. \vbox{             : 箱を縦に積む
6. \hrule height.1ex width.98\wd0 : 箱の定義（幅 0.98wd0，高さ 0.1ex の横線）
7. \kern.33ex         : 0.33ex の長さの空白を入れる（縦書きモードなので縦方向）
8. \hrule height.1ex width.98\wd0 : 箱の定義
9. \kern.45ex}}       : 0.45ex の長さの空白を入れる
```

結局、文字 Y(の箱) に横線 2 本 (が描かれた箱) を負の空白によって重ねあわせているのです。文字 Y の大きさに対する比で線の大きさが定義されていますから、フォントの大きさを変えても形が崩れません。

footnotesize の ¥　Large の ¥　huge の ¥

書体を変えると、ちょっと旨くない場合もあるようです。

tt の ¥　**bf の ¥**　*it* の ¥　sf の ¥

練習 2

では、\LaTeX の定義を解読してください。latext.tex にありますから、

```
grep \LaTeX /usr/local/lib/tex/macros/latex.tex -5 | less
```

とでもして探してください。

```
% THE \LaTeX LOGO IS DEFINED HERE.
%
\def\LaTeX{{\rm L\kern-.36em\raise.3ex\hbox{\sc a}\kern-.15em
    T\kern-.1667em\lower.7ex\hbox{E}\kern-.125emX}}
```

この定義による字形はどんな場合に崩れるでしょうか?

練習 3

円記号を文字 Y と\rule[lift]{width}{height} でマクロ定義してみましょう。例えば

```
\def\Nyen{\setbox0=\hbox{Y}Y\kern-.85\wd0\rule[.78ex]{.70\wd0}{.1ex}%
\kern-.70\wd0\rule[.40ex]{.70\wd0}{.1ex}}
```

のような感じです。　　出力例： ¥(\yen) と ¥(\Nyen)

練習 4

内積 (Inner product) の記号を定義してみましょう。シュルティス [4, 邦訳 p.61] には\stackrel を使った方法が載っていますが、ここでは、\raise と\hbox を使って定義してみてください。表示は次のようになります。　　出力例：A•B **A•B** **A•B**

1.2　引数がある場合

引数を使うともっと利用範囲が広がります。

置き換えマクロ：引数あり

```
\def\マクロ名#1#2...{内容 (#1,#2,... を含む)}
```

例えば

```
\def\MaxwDist#1{\int\!f(v)#1\,dv}
```

と定義して、

```
$$\MaxwDist{v} \quad \quad\MaxwDist{(V(x)-K)} \quad \MaxwDist{{\bf 被積分関数}}$$
```

と#1 にいろいろな (実) 引数を渡すと、

$$\int f(v)v\,dv \quad \int f(v)(V(x)-K)\,dv \quad \int f(v)\ \text{被積分関数}\ dv$$

のような表示が得られます。

練習 5　次の表示を得るようなマクロ定義を完成させなさい。

- `\def\Lap{\frac{\partial^2}{\partial x^2}... }`

$$\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}$$

- `\def\Shrod#1{...\Lap\right}#1+V#1... }`　(#1 に`\mit\Psi(\mbox{\boldmath r})` を渡した結果。)

$$-\frac{\hbar^2}{2m}\left\{\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}\right\}\Psi(\mathrm{r})+V\Psi(\mathrm{r})=E\Psi(\mathrm{r})$$

- `\def\GaussInt#1#2#3{\int\!\!\!\!_ ... }`　(#1,#2,#3 に A, S', V を渡した結果。)

$$\iint_{\mathrm{S}'} A_n dS=\iiint_{\mathrm{V}} \mathrm{div}\boldsymbol{A}\,dV$$

`\def` では、引数中に段落モード (空行や`\par`) を入れられません。段落モードを引数に取りたい場合には `\long\def` で定義します。

練習 6

`\long\def` を使って段落を枠で囲むマクロ`\Frame#1#2` を定義してみなさい。#1 には枠の幅を入れるものとします。例えば

```
\Frame{0.4\textwidth}{段落 1\par 段落 2\par 段落 3}
```

のようにすると

段落 1 段落 2 段落 3

のような結果が得られるようにしたいのです。

2　フォントあれこれ

2.1　非標準フォントのロード

非標準フォントを使う場合には

フォントのロード

```
\newfont{\フォント名}{フォントメトリック [scaled 拡大率]}
```

によってフォントを読み込みます。\ フォント名は自分で任意の名前を定義します。フォントメトリックは ∗.tfm という名前のファイルで、通常は /usr/local/lib/tex/fonts にまとめられています。拡大して使いたい場合には' scaled 拡大率 (1000 が 1 倍) ' を設定します。例えば

```
\newfont{\Normu}{cmu10 scaled 1000}
```

によって、次のように 10pt の立体イタリック文字 (?) が使えます。

A quick brown fox jumps over the lazy dog.

2.1.1　日本語フォント

LINUX+JE では日本語のフォントは Zeit 形式のベクトルフォントを利用することになっています。従って、任意の大きさをロードできます。例えば

```
\newfont{\BBiggoth}{goth10 scaled 30000}
```

により、次に示すように 300pt の大きさのゴシックを扱えます。

図 1　300pt のゴシック文字の出力例

2.2　フォントの一覧

記号等を\char"23 のように 16 進数で、または\char'043 のように 8 進数で指定するには、フォントの一覧があると便利です。TEX で (LATEX では旨く行きません) testfont.tex を対話的にコンパイルすると一覧表や見本が得られます。例えば次に示すような感じの表が得られます。

表 1　cmu10 フォントの一覧

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	″0x
´01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
´02x	ı	ȷ	`	´	ˇ	˘	¯	°	″1x
´03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
´04x	‘	!	”	#	£	%	&	’	″2x
´05x	(	)	*	+	,	-	.	/	
´06x	0	1	2	3	4	5	6	7	″3x
´07x	8	9	:	;	¡	=	¿	?	
´10x	@	A	B	C	D	E	F	G	″4x
´11x	H	I	J	K	L	M	N	O	
´12x	P	Q	R	S	T	U	V	W	″5x
´13x	X	Y	Z	[	“	]	^	˙	
´14x	‘	a	b	c	d	e	f	g	″6x
´15x	h	i	j	k	l	m	n	o	
´16x	p	q	r	s	t	u	v	w	″7x
´17x	x	y	z	–	—	″	~	¨	
	″8	″9	″A	″B	″C	″D	″E	″F	

2.2.1　フォントを使った枠組み

dingbat という名前のフォントを使った飾り罫線を紹介します。まず、フォント一覧を見て下さい。

表 2　飾り罫線用フォント：dingbat の一覧

´102	´103	´104	´105	´106	´107
´110	´111	´112	´113	´114	´115
´116	´117	´123	´132	´141	´142
´143	´144	´145	´146	´147	´150

これの METAFONT のソースに飾り罫線枠で文章を囲むマクロが付属していましたので、ちょっと手を加えました (オリジナルでは漢字と相性が悪いようでした)。

```
\newfont{\ding}{dingbat scaled 1000}

\long\def\RRamec#1{
\setbox0=\hbox{\ding\char'141}
\newdimen\charwidth \charwidth=\wd0
\newdimen\charheight \charheight=\ht0
\def\updoublerulefill{\xleaders\hbox to 0.6\charwidth
            {\hss\ding\char'142\hss}\hfill}
\def\downdoublerulefill{\xleaders\hbox to 0.6\charwidth
            {\hss\ding\char'147\hss}\hfill}
\def\leftdoublerulefill{\xleaders\vbox to 0.6\charheight
         {\vss\hbox{\ding\char'144}\vss}\vfill}
\def\rightdoublerulefill{\xleaders\vbox to 0.6\charheight
        {\vss\hbox{\ding\char'145}\vss}\vfill}
\newdimen\Width \newdimen\Height
\newdimen\Widthn \newdimen\Heightn
\setbox0=\vbox{#1}
  \Height=\ht0 \advance\Height\dp0
  \Width=\wd0
  \Widthn=\Width  \advance\Widthn 2.1\charwidth
  \vbox{%
  \hbox to\Widthn{\ding\char'141\updoublerulefill\ding\char'143}
  \hbox to\Width{
    \hskip-2.3pt\vbox to\Height{\leftdoublerulefill}%
    \hfill\box0\hskip22.1pt\hfill%
    \vbox to\Height{\rightdoublerulefill}%
    }
  \hbox to\Widthn{\ding\char'146\downdoublerulefill\ding\char'150}}
}
```

使い方は簡単です。\fbox 等と同様に、

```
\hfill\RRamec{\hsize=10cm \いろは \いろは \いろは \いろは}\hfill{}
```

と文章を囲みます。\hsize で幅を指定しなければ、横幅一杯になります。

飾り罫線枠の出力例

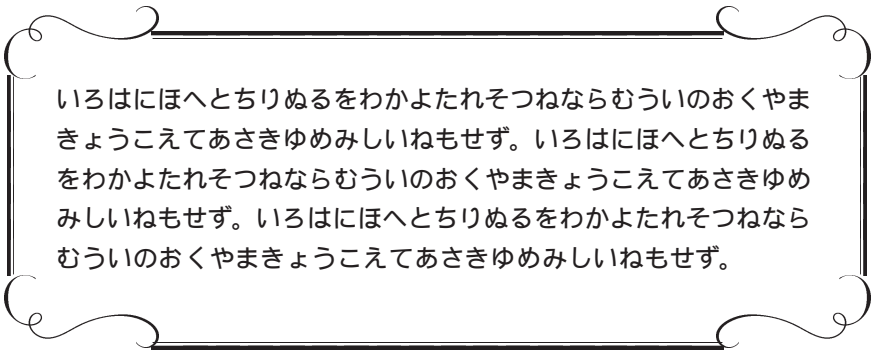
いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

このようなマクロの中身を全部知る必要はありません。罫線の種類を変えたければそれらしき場所のみ変更

すればいいのです。dingbat のフォント一覧と睨み合わせて、飾り罫線に使う文字の指定は\ding\char'###のところであるとすぐにわかるでしょう。四隅の文字は 20,26 行で、横罫線は 5 ~ 8 行、縦罫線は 9 ~ 12 行で定義されています。ここを適当に変えてしまえば新しい飾り罫線枠ができるはずです。

練習 7

\RRamec のマクロ定義を基にして、四隅の文字設定を変えて次のような飾り罫線枠を実現しなさい。



3　様々なスタイルファイルの利用

練習 8

jtwocolumn.sty を使って、8 ページのようなレイアウトを作成しなさい。

練習 9

fancybox.sty を使って、9,10 ページのようなレイアウトを作成しなさい。

薄膜の結晶成長のアニーリング温度依存性

東京電機大学工学部　　松田七美男, 松田 彩

1. はじめに

A quick brown fox jumps over the lazy dog.

$$\iint_{\text{面}} Q_n dS = \iiint_{\text{空間}} \mathrm{div}\boldsymbol{Q}\, dV \qquad (1)$$

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

2. 装置

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。いろはにほへとちりぬるをわかよたれそつねならむうい

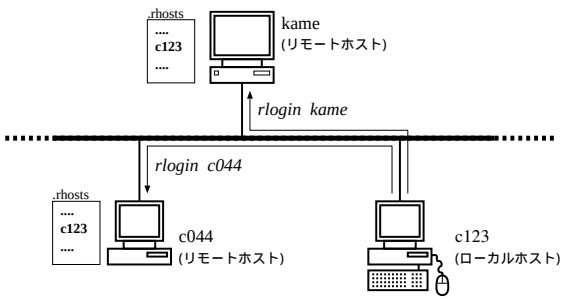


図 2　実験装置の図

のおくやまきょうこえてあさきゆめみしいねもせず。

3. 結果と考察

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

表 3　結晶粒径の温度依存性

温度 [K]	結晶粒径 [μm]
304	1.2
423	2.3
634	3.6

A quick brown fox jumps over the lazy dog. いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

4. まとめ

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

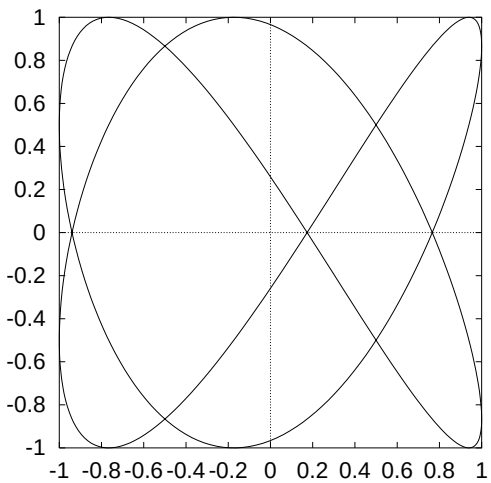


図 3　実験結果

- A quick brown fox jumps over the lazy dog.
- いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

参考文献

[1] Donald E. Knuth, *The TEXbook*, (Addison-Wesley Publishing Co., 1984). 邦訳：『TEX ブック』 鷺谷好輝 訳　斎藤信男 監修, (アスキー出版局, 1989).

[2] 磯崎秀樹,『LATEX 自由自在』Computer Today ライブラリ– 5, (サイエンス社, 1992).

[3] 岩熊哲夫, 古川徹生,『LATEX のマクロやスタイルファイルの利用』(東北大学の FTP サーバー ftp://ftp.touhoku.ac.jp/pub/TeX/latex-styles/bear_collections/ にあるフリー配布版, Version 2.15, 1994)

[4] J. Kenneth Shultis, *LATEXNotes: Practical Tips for Preparing Technical Documents*, (Prentice Hall, New York, 1994). 邦訳：『LATEX 実用ハンドブック』河原林美子, 内山厚子 訳　鷺谷好輝, 阿瀬はる美 監修, (プレティンスホール出版, 1995).

[5] 藤田眞作,『LATEX マクロの八衢』Higher Education Computer Series 15, (アジソン・ウェスレイ・パブリッシャーズ・ジャパン, 1995).

[6] 生田誠三,『LATEX 文典』(朝倉書店, 1996).

禁
煙

真空

4　レイアウトの情報

手仕事によるレイアウト調整はなるべく避けるのが LaTeX の使い方の原則です。しかし、中級ともなると、少しいじりたいと思うでしょうし、実際にどのように設定されているのか知りたいでしょう (こうして泥沼に落ち込んでいくのかもしれない...)。生田 [6, p.256] には標準設定値が載っていますが、全ての場合を尽くしていません。そこで、現在の設定値を表示するマクロを定義してみましょう。

4.1　長さの値表示：\the

伸縮する長さ (グルー) は一般的に次のように定義されています。

```
\newskip\Skipaa
\Skipaa 4.0pt plus 1.5pt minus 0.5pt あるいは
\Skipaa=4.0pt plus 1.5pt minus 0.5pt
```

最初の宣言は新しいグルーを使うための宣言です。この値を表示するには\the を使います。すなわち、\the\Skipaa とすると 4.0pt plus 1.5pt minus 0.5pt と値が表示されます。名前を補って表示するよう

```
\def\DispSkipaa{$\backslash$Skipaa=\the\Skipaa}
```

とでもマクロ定義すれば、文掌中で \DispSkipaa により

\Skipaa=4.0pt plus 1.5pt minus 0.5pt

が出力されます。この調子で、\baselineskip 等を表示するマクロを定義すればいいのです。

練習 10

\baselineskip を表示するマクロを定義して、フォントの大きさや環境を変えた場合に値がどのように変わるか調べてみなさい。単純なものでは

\baselineskip=15.0pt　small は \baselineskip=11.0pt　large は \baselineskip=17.0pt

のようになり見苦しいですから、次のように表示フォントサイズを常に\small に固定するようマクロを改良してみてください。

Large でも表示は \baselineskip=21.0pt　normalsize でも表示は \baselineskip=15.0pt
tiny でも表示は \baselineskip=6.0pt

ページレイアウトを表形式で出力するためのマクロを紹介しておきます。他のグルーは、このように直接 tabular 環境に入れる訳には行きません (理由は至極簡単ですから考えてください)。

```
\newskip\Sk
\def\DispSk#1{{\small\tt #1=\the\Sk}}
\newdimen\Sd

\long\def\SkipPage{
\begin{tabular}[t]{llll}
%\noalign{\hrule height 1pt}
\Sk=\the\textwidth \DispSk{textwidth} &
\Sk=\the\textheight \DispSk{textheight} &
\Sk=\the\oddsidemargin \DispSk{oddsidemargin} &
\Sk=\the\evensidemargin \DispSk{evensidemargin} \\
\Sk=\the\topmargin \DispSk{topmargin} &
\Sk=\the\headsep \DispSk{headsep} &
\Sk=\the\headheight \DispSk{headheight} &
%\Sk=\the\headrulewidth \DispSk{headrulewidth} \\
\Sk=\the\topskip \DispSk{topskip} &
```

```
\Sk=\the\voffset \DispSk{voffset} &
\Sk=\the\hoffset \DispSk{hoffset} \\
\Sk=\the\footnotesep \DispSk{footnotesep} &
\Sk=\the\footheight \DispSk{footheight} &
\Sk=\the\footskip \DispSk{footskip} &
%\Sk=\the\footrulewidth \DispSk{footrulewidth}\\
\Sk=\the\marginparsep \DispSk{marginparsep} &
\Sk=\the\marginparwidth \DispSk{marginparwidth} &
\Sk=\the\marginparpush \DispSk{marginparpush} \\
\Sk=\the\columnsep \DispSk{columnsep} &
\Sk=\the\columnwidth \DispSk{columnwidth} &
\Sk=\the\columnseprule \DispSk{columnseprule}
%\noalign {\hrule height 1pt}
\end{tabular}
}
```

文章中で\SkipPage とすると、ページレイアウト情報が表示されます。

textwidth=455.24408pt　textheight=682.86613pt　oddsidemargin=0.0pt　evensidemargin=0.0pt
topmargin=-28.45274pt　headsep=25.0pt　headheight=17.0pt

marginparsep=10.0pt　marginparwidth=60.0pt　marginparpush=5.0pt
columnsep=30.0pt　columnwidth=455.24408pt　columnseprule=0.0pt

5　リスト処理

区切り記号のないリスト処理を使うコマンドとして\@tfor があります。これを練習をしてみましょう。

練習 11

引数に文字列を与えて 1 文字毎に枠を表示するマクロを考えてみなさい。文字の箱の高さや幅や深さが視覚的に理解できます。基準線 (baseline) を付けた例を示します。しかしこれでは、グルーが無視されてしまい微妙な文字間の調整が反映されず、実際の組み上がりとは異なります。

baseline 1 2 4 5 6 7 8 9 0 a b c d e f g h i j k 愛 だ ろ 愛

baseline 1 2 4 5 6 7 8 9 0 a b c d e f g h i j k 愛 だ ろ 愛

TEXbook[1, p.98] に同じ問題があり、TEX での解答もありますが、面倒です。筆者が考えた正確でないマクロは、次のようなもので、上に述べた重大な欠点がありますが、雰囲気は掴めるでしょう?

```
\makeatletter
\def\EveryFrame#1{{\normalsize\bf baseline}\rule{10pt}{0.1pt}%
\fboxrule 0.1pt \fboxsep 0pt%
\@tfor\member:=#1\do{\fbox{\member}}%
\rule{20pt}{0.1pt}}
\makeatother
```

6　データファイルからの直接の作表

4 カラムの空白で区切ったデータが保存されているファイルがあるとします。これを LaTeX の表にするには、**awk** や **perl** といったテキストを扱うスクリプト言語を用いるのが簡単です。awk などで sample.dat から sample.tex を生成し、\input{sample.tex}を行えばいいのです。これでも、awk で前処理が必要ですね。そこで、

練習 12

磯崎 [2, p.139] を参考にして、データファイル名前を引数に取って表を自動作成する LaTeX マクロを考えてみなさい。出力例を示します。

表 4　データファイルを読み込んで自動作成した表

E_p [eV]	I_0 [nA]	I_{50} [nA]	δ
100	1.2	-0.6	1.12
200	2.3	-0.9	1.34
300	3.2	-1.1	1.56
400	4.5	-1.0	1.23
500	5.0	+0.0	1.00

これもかなり面倒です。どうもアスキー日本語 TeX では、\newread や\read... to... は日本語のコントロールシーケンスを理解しないようです。筆者が捻り出したマクロは以下のようなものですが、これは結構旨く行くようです。全角の数字を半角数字にすると失敗します。

```
\def\第 1 欄#1 #2 #3 #4;{#1}
\def\第 2 欄#1 #2 #3 #4;{#2}
\def\第 3 欄#1 #2 #3 #4;{#3}
\def\第 4 欄#1 #2 #3 #4;{#4}
\newif\if 読み込み
\newread\DF
\def\SEE データ表#1{\leavevmode\vbox{%
\openin\DF=#1
\endlinechar`;
\def\空行{;}%
\読み込み true%
\hrule
\hbox {%
\vrule\strut%
\hbox to 2cm{\hss$E_p$ [eV]\hss}\vrule
\hbox to 2cm{\hss$I_0$ [nA]\hss}\vrule
\hbox to 2cm{\hss$I_{50}$ [nA]\hss}\vrule
\hbox to 2cm{\hss$\delta$\hss}\vrule
}
\hrule
\loop
  \read\DF to\D
  \ifx\D\空行\else
  \edef\エネルギー{\expandafter\第 1 欄\D}%
  \edef\電流 0 {\expandafter\第 2 欄\D}%
  \edef\電流 5  0 {\expandafter\第 3 欄\D}%
  \edef\係数{\expandafter\第 4 欄\D}%
  \hbox {%
  \vrule\strut%
  \hbox to 2cm{\hss\エネルギー\hss}\vrule
  \hbox to 2cm{\hss\電流 0 \hss}\vrule
  \hbox to 2cm{\hss\電流 5  0 \hss}\vrule
  \hbox to 2cm{\hss\係数\hss}\vrule
  }
  \fi
```

```
    \ifeof\DF\読み込み false\fi
    \if 読み込み\repeat
    \hrule}
  \closein\DF
  }
```

参考文献

[1] Donald E. Knuth, *The TEXbook*, (Addison-Wesley Publishing Co., 1984). 邦訳:『TEX ブック』 鷺谷好輝 訳　斎藤信男 監修, (アスキー出版局, 1989).

[2] 磯崎秀樹, 『LATEX 自由自在』Computer Today ライブラリ– 5, (サイエンス社, 1992).

[3] 岩熊哲夫, 古川徹生, 『LATEX のマクロやスタイルファイルの利用』(東北大学の FTP サーバー ftp://ftp.touhoku.ac.jp/pub/TeX/latex-styles/bear_collections/ にあるフリー配布版, Version 2.15, 1994)

[4] J. Kenneth Shultis, *LATEXNotes: Practical Tips for Preparing Technical Documents*, (Prentice Hall, New York, 1994). 邦訳:『LATEX 実用ハンドブック』 河原林美子, 内山厚子 訳　鷺谷好輝, 阿瀬はる美 監修, (プレティンスホール出版, 1995).

[5] 藤田眞作, 『LATEX マクロの八衢』Higher Education Computer Series 15, (アジソン・ウェスレイ・パブリッシャーズ・ジャパン, 1995).

[6] 生田誠三, 『LATEX 文典』(朝倉書店, 1996).

[7] 奥村晴彦, 『[改定版]LATEX 2ε美文書作成入門』(技術評論社, 2000).

[8] 中野賢, 『日本語 LATEX 2ε ブック』(アスキー, 1998).