







## 11.11 TeX の行・段落整形

TeX の整形は、一行別で完結するのではなく、デメリット ( $d$ ) という評価値をもつに段落全体にわたっておこなう。詳しくは、拙著「`LaTeX` 本づくりの八幡」を参照。その中で、`\tracingparagrapas` 命令を用いる追跡方法についても詳しく述べられている。次に掲げるのは、その要約である。

英文の場合、行の分割箇所は、グルー類に属するノードから選ぶ<sup>18</sup>。改行不能のノードが行末にくる場合は、グルーを縮めてその行に収めるか、グルーを延ばして次行に送るか、英単語ではハイフネーションで分割するかの判断をする。この判断は、デメリット ( $d$ ) という評価値を計算する。

$$d = \begin{cases} (\ell + b)^2 + p^2, & 0 \leq p < 10000 \text{ のとき} \\ (\ell + b)^2 - p^2, & -10000 \leq p < 0 \text{ のとき} \\ (\ell + b)^2 + p^2, & p < -10000 \text{ のとき} \end{cases}$$

`\lignp` と `\linepenalty` と呼ばれる量で通常 10 に設定される。 $b$  は悪評度 (badness) で、行整形のために伸び縮みしたグルーから計算されるが、おおむね三乗のオーダーの値である。 $p$  は分割可能点に与えられた罰金の値である。この値によって計算したデメリットを、段落にわたって総計して、最小のものを選ばうのがマルトリズムになる。

和文の場合、TeX も TeX の行分割アルトリズムを利用している。このとき和文と英文の混植 (混合植字) など和文特有の事情に対処するために、いくつかの規約 (和字・和字間のグルー、和字・英字間のグルーなど) を追加している (前述)。

## 11.13 TeX の禁則処理機能

TeX の禁則処理プログラムの詳細は、拙著「`LaTeX` 本づくりの八幡」を参照。要約を次に示す。

行頭禁則 行頭禁則の文字は、その文字の直前で行分割の処理が行われないようにするために、`\prebreakpenalty` に大きな値を設定する。この設定は、`kinsoku.tex` の中で行われる<sup>19</sup>。たとえば、

```
\prebreakpenalty' =10000 %和字 (半角) の字の罰金  
\prebreakpenalty' =10000 %和字の字の罰金  
\prebreakpenalty' =5000 %和字 (全角) の字の罰金  
\prebreakpenalty\jis"212B=5000 %リード直前
```

なすの字のリード直前をとり、変更も可能である。

行末の禁則処理 行末禁則の文字は、`\postbreakpenalty` に大きな値を設定して、その文字の直後に行分割の処理が行われないようにする。

段落ウェアウ処理 TeX では、段落末尾の行が 1 文字にならないうちに整形する<sup>20</sup>。このように禁則処理を、段落ウェアウ処理と呼ぶ。これを実現するために、`\charwidowpenalty` が、該行箇所<sup>21</sup>に自動的に挿入される。この罰金の値は `jplain.tex` の中で 500 に設定されているが、変更も可能である。

<sup>18</sup> 行整形の 1 つの候補は、分割箇所を改行直前の木構造 (層) を用いて選ぶことである。このように分岐点を一歩下ろぐ。





- 1 ベタ組は金科玉条か。
- 2 「調整箇所をできるだけ少なく」という調整原理を維持すべきか。
- 3 前項とほとんど同じことになるが、調整の優先順位を固定的に考えることは非。優先順位の高いところで調整ができればそれで終わりとする方法の吟味。
- 4 句読点のぶら下げは本当に必要なのか。

### 三・二 コンピューター組版によって可能になったこと

「コンピューター組版によって、技術的に可能になったことは何かを考えると、次の二点になる。」

- 1 文字と文字の間隔を自由に調整できるようになったこと。
- 2 調整のための演算が容易にできるようになったこと。

この進歩を反映するように、組版ルールを吟味することが必要である。右にあげたことからは、「和字をベタ組することを原則としなくてもよい」ということを意味する。さらに敷衍すれば、「和字<sup>22</sup>・和字の間を積極的に行の調整に使うことができるようになった」と解釈することができる。これは、「調整箇所をできるだけ少なく」という調整、さらには、「優先順位の高いところで調整ができればそれで終わり」とす調整のくびきから自由になることを意味する。これらを踏まえて、本報告では、次のような立場をとることにする。

- 1 ベタ組の桎梏を取り除く。
  - 2 すべての調整可能な箇所<sup>23</sup>にわたって、優先順位を考慮した重みづけ調整をおこなう。
  - 3 行末の約物の取り扱いについて、一つの約物に対して、単一の仕上りとする。本報告では、単一行末揃えの原則という言葉で言及する。
  - 4 句読点のぶら下げをおこなうかどうかは、文書の内容に応じてきめる。ただし、筆者は、ぶら下げをおこなわない方法をデフォルトとする立場である。
- 次節から、これらの事項の細部を例をあげながら述べてゆく。

### 三・三 版面をまもることの意味

縦組の組版において、版面をまもることの意味を考えてみよう<sup>24</sup>。

版面の左右 まず、版面の左右について考える。見開きで左右対称になるようにしたいという要請と本文の中心線を裏表で一致させたいという要請とをみたすことが肝要である。見開き二ページを単位にして考えるとわかるように、小口<sup>25</sup>に柱を出力するときに、制約がもっとも厳しい。右ページ（縦組のときは偶数ページ）の右側にくるルビ、割注の右側などは、版面から飛出しても可とする。左ページ（縦組のときは奇数ページ）の左側にくるものは少ないが、漢文の返り点などがある。これも、版面から飛出しても可とする。本報告では、左右揃えの原則という言葉で言及する。

版面の上部 版面の上は、字余りのルビなども飛出してはならないとするのが通例である。もちろん、特殊な効果をねらうときは別である。版面の上部に関しては、段落行頭の括弧と折り返し行頭の括弧についての相異はある。とくに、

22 このことにより、「通常処理」と「例外処理」の関係が変わる。たとえば、行頭にカギ括弧がきて、そのりの文字は漢字またはかなで構成されている行の調整を考えると、半端の二分を処理するために文字間を開ける必要がある。ベタ組原則（ベタ組が「通常処理」の立場からは「この処理は「例外処理」となる。小書注や割注などを採用する場合など、この種のことはいくさんあるので、ベタ組原則の立場では、例外処理が多発する。一方、本報告で述べるベタ組離脱の立場からは、最初の例の処理は「通常処理」であり、ベタ組もまた文字間をopenに開ける「通常処理」である。小書注や割注（この場合は半端の留保はあるが）の処理でさえ、「通常処理」である。いいかえれば、ベタ組原則の立場からの「調整」は例外の処理である。ベタ組離脱の立場からの「調整」は、すべて正常の処理である。「この相異は、単なることは遊びではなく、深い意味をもっている。」

23 「このことにより、調整箇所をできるだけ少なく」という調整原理を維持すべきか。前項とほとんど同じことになるが、調整の優先順位を固定的に考えることは非。優先順位の高いところで調整ができればそれで終わりとする方法の吟味。

24 「このことにより、調整箇所をできるだけ少なく」という調整原理を維持すべきか。前項とほとんど同じことになるが、調整の優先順位を固定的に考えることは非。優先順位の高いところで調整ができればそれで終わりとする方法の吟味。

25 このことにより、調整箇所をできるだけ少なく」という調整原理を維持すべきか。前項とほとんど同じことになるが、調整の優先順位を固定的に考えることは非。優先順位の高いところで調整ができればそれで終わりとする方法の吟味。

26 段落の途中の行の先頭。



ファイルに禁則でつづき、本報告では、固定行送りを重視して、禁則をはずして、ウイドウ行の禁則をはずすという、ページの最後の行を「のページに禁則」が必要になる。「この報告では、ページの」がを一行分開くことは、ページの最後の行を「のページに禁則」部分で別行立て「この」行を繰くの表現は、TeX の「の」は、前ページの行間を均等に開くか、段落の間では半

るのが自然である。<sup>28</sup>

### 三・四 句読点細見

#### 句読点の形態的考察

従来、ぶら下げは、句読点のみに適用されてきた。ここで、は、「ぶら下げに関して、句読点だけを他の約物(行末禁則のかかるもの)とは別扱いにするという理由があるか」ということを考える。これは、句読点という概念的な考察だけでなく、句読点の形態的な考察が必要になる。これは、「句読点を句読点たらしめているものはなにか」という問いの答えをあたえることとなる。

一般に句読点は、縦幅半角(0.5zw)で、半角の空白が付属するとされる。縦幅半角(0.5zw)というのは、あくまで句読点の概念上の大きさである。句読点を実際に囲む最小限の枠を出力するつもりで見ると、実際にはもっと小さいことがわかる。筆者の $\text{P}^{\text{T}}\text{E}^{\text{X}}$ のシステムでは、0.27zw(四分より若干大きい)の枠で句読点を囲むことができる(ただし、句読点の縦幅などの具体的値はフォントのデザインによって異なる可能性があるが、異なったとしても議論の大筋はそのまま通用する)。句読点は、 $0.27zw \times 0.27zw$ の正方形の領域に収まる。

句点	0.5zw	0.27zw	0.27zw
読点	0.5zw	0.27zw	0.27zw
丸括弧	0.5zw	0.27zw	0.27zw
カギ	0.5zw	0.27zw	0.27zw
一重カギ	0.5zw	0.37zw	0.37zw

比較のため、終わり括弧類の実測値も並べて掲げた。丸括弧とカギ括弧は、句読点と同じく縦幅0.27zwであるが、横幅は全角である。一重カギ括弧は、縦幅0.37zw、全角の横幅となっている。句読点も終わり括弧類も「本文をその地点で区切る」という働きをまっとうする。形態的にみると、句読点の占める領域(0.27zw×0.27zw)が終わり括弧類の占める領域(0.27zw×1zw または 0.37zw×1zw)に比較して、圧倒的に小さいという点である。要するに、形態的にみた場合、句読点は区切る力が弱い。いいかえれば、そのまますたで(縦幅0.27zwで)組んでゆくと本文に紛れてしまうという点である。形態的に貧弱な句読点による区切りを、視覚的に確保するための仕掛けが次の二つである。

- 1 概念的に半角とみなすこと
- 2 半角の空白を付属させること

このことにより、形態的な区切る力の弱さをカバーすることができる。この二つの仕掛けが、「句読点を句読点たらしめているもの」にほかならない。句読点のつづる半角を詰めるような行調整はおこなわないという組版ルールは、句読点の形態的な弱さをあらわにしないための工夫であるともみさせる。

#### ぶら下げの論理

このようにみると、句読点のぶら下げは、形態的な弱さを逆手にとった方法であることがわかる。弱さをカバーするために、上記の二つの仕掛けには、版面の下部の余白が充てられる。前項から饅々述べ

てきたことは、要するに、「句読点は見かけ上小さいので、句読点をぶら下げにしてあまり目立たない」ということを、麗々しく言いかえたにすぎない。

ところで、「句読点以外は絶対にぶら下げにしない」という組版ルールについてもついでに吟味しておこう。丸括弧やカギ括弧は、縦幅だけについていえば句読点とほぼ同じである。これらの横幅は全角であり句読点に比べて大きく目立つのでぶら下げにしないといわれれば、そのようにも感ずる。しかし、本当にそうか、確かめてみる。

欄外に、句読点とともにカギ括弧もぶら下げにした例をあげる。例文中、

句読点と区別せず、ほかの約物  
鉤括弧をぶら下げると「可」  
は、  
目明のせいと感ずるのの「可」  
だ、  
もちろんだと感ずると野言で  
せるのだらうが。

（文章もおかしいがうとされた）。ここでは、大カギ括弧を使っているが、小カギ括弧を用いると占める領域は小さくなる。次の（参考一）にはカギ括弧をぶら下げにしないものをあげる。むしろ、（参考一）は、ぶら下げをおこなわない場合である。「の」三つを比較して、カギ括弧のぶら下げは、はたして奇異に感じられるであらうが。

版面下部の漢字やかなの並び

句読点と区別せず、ほかの約物  
鉤括弧をぶら下げると「可」  
は、  
目明のせいと感ずるのの「可」  
だ、  
もちろんだと感ずると野言で  
せるのだらうが。

句読点と区別せず、ほかの約物  
鉤括弧をぶら下げると「可」  
は、  
目明のせいと感ずるのの「可」  
だ、  
もちろんだと感ずると野言で  
せるのだらうが。

のみに着目する（これが句読点のぶら下げをおこなう大きな理由である）と、カギ括弧をぶら下げにしたほうが、横にならなびるともいえる（物「否」「る」に着目）。<理屈に近いこ

とを承知でいうならば、この着目事項を変えないことにすると、カギ括弧のぶら下げにしない（参考一）は、行末の漢字とかながむしろ凸凹しているといえる。一方、ぶら下げをまつたくおこなわない（参考二）は、最初の例と同様に版面下部の漢字やかなが並んでくる。

以上、いろいろと検討してみたが、「ぶら下げに関して、句読点だけを他の約物とは別扱いにする」という積極的な理由は筆者には見つからない。「句読点は見かけ上小さいので、句読点をぶら下げにしてもあまり目立たない」ということは、「句読点はぶら下げにしないで目立たない」ところもある。したがって、ぶら下げありの論理は、そのままぶら下げなしの論理にかわってしまふ。ぶら下げあり・なしの議論は、煎じ詰めれば、好みの問題とということになり、いずれを信ずるかという宗教戦争に似たものとなる。

#### 四 単一行末揃えの原則

##### 四・一 ぶら下げあり・なし

###### 二つの選択肢

行末揃え（できるかぎり版面の下部をそろえる）を原則として、さらにそれぞれの約物には、行末で単一の振る舞いしか許さないようにしよう（単一行末揃えの原則）。このような方針をたてると、行末にきた約物は二分として、うしろの空白を除いたうえで、行末尾に揃えることになる。このとき、句読点を他の約物と区別して特別扱いをし、ぶら下げを許すとすると、次の二つの方

29 このページは、本報告の「5.2.3 組版」で

ちいた組版格子(グリッド)を出力した  
サンプルである。組版作業の途中は、原  
稿全体でこのみづな形態で組版格子が  
出されていく。最終版では消去してい

30 「1」定義する。単一行未揃えによる

り下す。は、鈴木一誌、前田年周  
向井尚、ハ、明解リニエーターのため  
印刷ガイドブック(エムエム実業社、光  
社)一九九〇 Part 1)明解日本語文  
字組版の「強いから下す」とは、

31 「1」定義する。単一行未揃えによる

り下す。は、頭注の印刷文書の

「強いから下す」とは、

法が残る。<sup>29</sup>

30  
「強いから下す」とは、  
「強いから下す」とは、

1  
「強いから下す」とは、

「強いから下す」とは、  
「強いから下す」とは、





たがって、段落の先頭は必ず規定の位置（本報告の場合は字下げされた位置）にきて狂いが無い。

## 五 左右揃えの原則

### 五・一 行間要素のみ出し

`pdfTeX` は標準の傍点をつけるための `\bou` 命令が使える。この命令は版面右側へののみだしができるようになっていて、直前に示すように、傍点がついたまま自動的に改行されるので便利である。

```
欄外の例の傍点は、\boudashionのためと入力したものである。この命令も、実装の方法は異なるが、やはり、「幅 0pt の箱」のファイアコマンドを使うことである。
```

筆者が作成した `furikana` パッケージ（中シキル）および `furiknkt` パッケージ（肩シキル）では、`\kana` 命令のみが使えるようになっていて、`\jw` の命令は、`テノケル` では、`ル` が版面からはみ出さないようにになっている（行の中央線が移動する）。`ル` を版面からはみ出すようにできるスイッチが用意されている。次のように書くのである。

```
\kanadashion
\kanakdashion
```

```
PREMedia, Part 15, 11001年六月、四九ページに示された「坊ちゃん」の組版で、furikana パッケージ（中シキル）および furiknkt パッケージ（肩シキル）を読み込んだ pdfTeX で検証したものである。\jw と同じく同様、和字-和字間は \kanjiskip=0pt plus .125zw minus 0.03125zw と設定している。和字-欧字間は \kanjiskip=.25zw plus .25zw minus .0625zw と設定している。右にあげたスイッチを宣言したのが、\Kanaktt、\Kanactt、\#、\あらし などを入力する。次のような出力結果がえられる。
```

```
この例にお  
いても、句  
読点をすべ  
をぶらすべ  
命令（、。お  
よび、）に  
置き換えて  
いる。その  
わがな、こ  
の、句読点  
はぶらすべ
```

の位置にしている。英語の `right` は、次の行に送られている。これにともなって「あ」が一行めに送られ、一行めが延ばし処理で組まれる。

<sup>38</sup> このページに、傍点を実際に版面に施すのはみだした例をあげている。

## 五・二 高々 (横箱) Opt の箱

前述のルビを版面からほみ出すようにするため、スイッチとして `\kanadashion` を用い、`\kanaktdashion` を設けよう。これらも、横箱の高さを `Opt` での `\kanatemp` に揃いつつある。ただし、横箱の高さを `Opt` とするための指定の仕方は異なる。`\kanatemp` での命令を次のように定義する。次に掲げる定義は、`furikana` パッケージの命令の原型である。

```
\def\kanatemp#1#2{%
\setbox0=\hbox{\tiny #2}%
\setbox0=\hbox{\raise\dp0\box0}%
\leavevmode
\vbox{\hbox{\dp0=opt \ht0=opt\box0}%
\nointerlineskip\hbox{#1}}}%
```

`\setbox0` の、`数#2` を指定されたルビを `\tiny` の大きさに横箱として組版し、高々 `\dp0` を与えるためにも、一度横箱として組版する。これでルビの部分が完成する。これを `もじ`、縦箱 `\vbox` の中で、ルビの深々 `\dp0` と高々 `\ht0` を `Opt` として、親文字 `#1` を纏める。このように定義した命令を使い、`\kanatemp{地蔵菩薩}とじ` と指定して出力したものを次に示す。一語目は `\kana` 命令を使った比較例である。

`\kana` 命令を使った比較例

## 六 特殊な組版

### 六・一 漢文の訓点文

漢文の訓点文を組版するため、`sfkanbun.sty` を開発し、筆者のホームページで公開している。`\kundoku{田中つとむ( )}` などを入力するようにより、次のような出力がえられる。論語「公冶長論」より引用する。

`\kundoku` 命令は、これまで述べてきた幅・高々 `Opt` の箱のマイディアの応用である。句読点を送りがなの部分に潜り込んだ組版が可能になっている。また、版面の右側から横幅なしではみだすようになっている。 `\kundoku` 命令を繰り返しもちいなければならぬといつデメリットを補って余りある効果がえられる。このデメリットも入力エディター側の補完ソフトウェアで解決できる。

### 六・二 数 式

数式の組版は、`TeX` のもっとも得意とする分野である。縦組の書物でも、数式を組み入れる機会は今後ますますふえてくるだろう。`TeX` の数式組版については多くの参考書があるので、本報告ではごく簡単に触れるにとどめる。

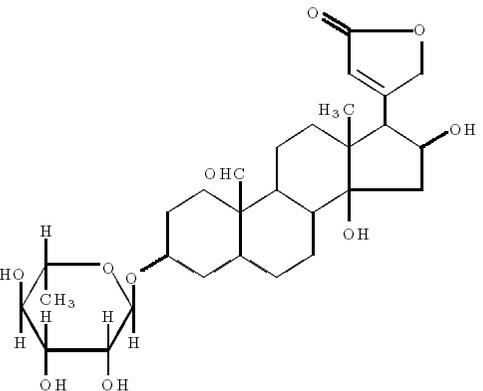
欄外に示した数式は、通常に組版した数式を縦横の幅 `Opt` の箱 (要するに点) に閉じ込めて出力したものである。この方法は、段落始めの欄外に、をおくやり方(前

39 この入力には、レイトウデータが補完されていないことに注意。出力では、送りがなの「ハク」に読点が置かれないこと。`TeX/LaTeX` 処理の「後」に、送りがなの `OK` 列の長さを自動的に正しい「読点」の出力位置を調整しているため、`TeX` をプログラム言語と称する近頃は、

述)と本質的には同じものである。最初にこの出力をつるには、若干の試行錯誤が必要である。しかし、一度手法が確立してしまつと、これをマクロ化することにより、それ以後は造作なく組版することができるようになる。なお、`TeX`組版の柔軟性を示すために、出力した数式は、故意に左側の欄外にも飛出させてある。

### 六・三 化学構造式

筆者は、化学構造式を描くためのパッケージとして`XYTEX`を開発して公開している。筆者のホームページより、パッケージ本体と`Al`が入手可能である。また、マニュアルを「`XYTEX—Typesetting Chemical Structural Formula`」(マシンウエスライ、一九九七)として刊行している。`XYTEX`と、`Al`の処理系で動くマクロ集である。`XYTEX`の命令を一定の文法で組み合わせれば、かなり複雑な構造式でも容易に描くことができる。欄外に示したのは、`adomitoxin`の構造式であり、次のような命令で描いたものである。



```
\steroid{{10}}=\lmoiety{OH};{{14}}=OH; %
{{13}}=\lmoiety{H}_{3}{C};{{16}}=OH; %
{{17}}=\fiveterov[e]{3=0}{4D=0;1=(y1)}; %
3=\lyl{3=0}{8= %
\pyranose{15b=(y1);15a=H;25b=H;25a=OH; %
35b=H;35a=OH;45b=H0; %
45a=H;55b=H;55a=CH}_{3}}}
```

そのうえで、縦横の幅`Opt`の箱(要する点)に閉じ込めて、欄外に出力したものである。`XYTEX`の組版は、構造式素片を組み合わせる活字時代の方法に概念的には近いものである。外部図形の取り込みによるものではない。

### 七行どり

#### 七・一 縦組における行どりの重要性

版面の左右をまもるほかに、地の文の行位置をページの裏表で揃えたいという要請もある。この要請を行送り固定の原則とよぶことにしよう。行送り固定は、`TeX`に限らず、日本語化したDTPソフトウェアにおいても、もともとむずかしい事項である。組版の作業性・経済性を重視して(要するに手を抜いて)、この要請を棚上げにし、版面の左右のみをまもることのみをおこなうことも多い。筆者は、この段階でも許容レベルであると判断している。これまでの拙著の組版も、プログラムのソースリストや例文などを多く含むので、行送り固定の原則を棚上げにし、版面の左右のみをまもることのみをおこなってきた。

しかしながら、行送り固定は`TeX/Al`で不可能な<sup>41</sup>ことではない。本報告では、行送り固定の実現に真っ向から取り組んでみる。

行送りが乱れる要因には、次のものがある。

- 1 引用などで、文字の大きさを変更した場合は行送りがその部分だけ変わる。これらの前後に伸縮する空白をおき、版面の左右のみまもるように調整する。
- 2 別行立て数式などを挿入する場合、その前後に伸縮する空白をおく。数式全体

40 行送り固定のために、桁組格子(クリップ)を画面出力して、それを手がかりで組版する方法がある。

41 本報告は組版の途中では桁組格子を出力して、桁組格子を消した上で最終出力をおこなっている方法をとった。

42 行送りの整数倍という表現は、厳密には正しくない。行送りの領域の大きさは、この文字の幅幅分を加える必要がある。

の大きさによって、一般には、行送りの整数倍にならない。  
3 (D)E<sub>TEX</sub>の見出しの出力は、\section命令などではない。このとき、行送り方向の半端を、見出しの前の伸縮空白で調整する。この調整は、一般には、行送りの整数倍にならない。

4 ならびに、見出しの泣き別れを防ぐための調整、ヘーシ冒頭のウイットウ行 (widow line) やヘーシ末尾のクラブ行 (club line) の処理など、行送りが乱れる。これらの行送りを乱す組版要素を、行送りの整数倍にならないに矯正すれば、このようになる。

#### 七・二 行送り命令

行送りを乱す組版要素を、行送りの整数倍に矯正する環境として、筆者は gyodori 環境を作成している。この環境を含んだ sfgyodor パッケージを公開する予定である。このパッケージを \usepackage{sfgyodori} として読み込めば、gyodori 環境が使えるようになる。たとえば、

```
\begin{gyodori}{2}
\fbx{この部分に三行送りしたまの}
\end{gyodori}
```

のように入力する。このソースリストの部分も gyodori 環境をもちいて、三行送りになっている。出力結果を次に示す。

**この部分は三行送りしたまの**

43 三行送りが、TEX のパラメーターは、行送りと行間を区別して、前の行の文字が重なった場合は行間を広げるというメカニズムが働く。

たしかに三行送りになっているのがわかる。gyodori 環境の中では、TEX の行送りに関するメカニズムを利用しつつ、\baselineskip、\lineskip、\lineskiplimit などのパラメーター類や、できあがった箱の高々・深々などを調整する。実際の組版作業においては、行送りの行数を適正に選択することにより、行送りの半端を吸収するようになる。

#### 七・三 別行立て数式の行送り

(D)E<sub>TEX</sub> では、別行立て数式を組版するためには、equation 環境や equarray 環境をもちいる。このらを gyorodi 環境で囲めば、行送りを自由なようにできる。数式を行送りした中央にもつてくるには、若干の調整が必要である。入力は次のようになる。

```
\begin{gyodori}{3}
\begin{equation}
\mathbf{c}_{-x} = \mathbf{a}_{s_{-x}}
= \{s_{-x}, s_{-x}^{-2}, \dots, s_{-x}^{-x},
s_{-x}^{-x+1}, s_{-x}^{-x+2}, \dots, s_{-x}^{-2x}\} (=D)\}.
\end{equation}
\vskip0.7cm%間隔
\end{gyodori}
```

出力結果は次のようになる。行送りの行数は、数式の出力結果をみて適正に塩梅する。

$$C_{xh} = S_x = \{S_x, S_x^2, \dots, S_x, S_x^{x+1}, S_x^{x+2}, \dots, S_x^{2x} (=I)\}. \quad (1)$$

右の行どりの部分は、スーシの始めにくるようになり、TEXの処理では、行送りの空白分は自動的に捨てられる。スーシの中間にきた場合とは処理が異なっていることに注意されたい。

自動的に行数を勘定して、行どりをおこなうjequation環境も作成して、sigyodorパッケージに格納してある。この環境を使うには、次のように入力する。

```
\begin{jequation}
\mathbf{C}_{xh} = \mathbf{S}_x = \{\mathbf{S}_x, \mathbf{S}_x^2, \dots, \mathbf{S}_x, \mathbf{S}_x^{x+1}, \mathbf{S}_x^{x+2}, \dots, \mathbf{S}_x^{2x} (=I)\}.
\end{jequation}
```

出力結果は次のようになる。行どりの行数が自動的にきめられる。この場合は、二行よりになっている。

$$C_{xh} = S_x = \{S_x, S_x^2, \dots, S_x, S_x^{x+1}, S_x^{x+2}, \dots, S_x^{2x} (=I)\}. \quad (2)$$

このように、TEXの組版の癖に注意しつつうまく利用するようにより、高度な組版が可能になる。いろいろなパラメーターが用意されているので、これらを最適に調整するところが重要である。

#### 七・四 別行見出しの行どり

(p)LaTeXでは、\sectionや\subsectionなどの見出し出力用の命令が用意されている。日本語では、見出しの前の行送り方向のゲルをおくようになっていることが多く、固定行送りの障害となるようがある。一般に、日本語組版では、行どりを中央に見出しをおくことが多いので、この形態への対処をする必要がある。このためには、前述のgyodor環境の考え方を応用して、\sectionや\subsectionなどを再定義すればよいことになる。実際には本報告では、\sectionを変更して、二行中央どりの見出しを出力するものにする。また、\subsectionでは、一行あけて一行の別行見出しを出力する方式に変更している。たとえば、この項の最初には、\subsection{別行見出しの行どり}と何行かおいて、自動的に行のよみな形式になっているようがわかる。また、項の通し番号も自動的に振られる。やむを得ず、\sectionや\subsectionが続くときは、その間の間隔を一行減らすものに工夫している。

#### 七・五 特殊な見出し

##### 二行より同行見出し

この段落の冒頭には、二行どりの同行見出しと呼ばれる形式で、見出しが出力されている。筆者は、この種の見出しを実現するdaijingパッケージを作成し公開している。筆者のホームページよりこのパッケージを手すれば、hngmdsi環境（ハンギング見出し環境）などが使え

44 英文では、版面をまとめるために行送りを変更するのは普通に行なわれる。よく見出しの前や段落の間を開いて版面を調整することが多い。LaTeXの標準半角の仕様に気づいて、日本語組版の固定行送りのためにはこのメカニズムを抑制することが、うまく組版するためのキーポイントである。



現在の tochn パッケージの行間の注番号は、版面右側へのはみだしには対応していない。行間の注番号を出力する命令の定義を tochn パッケージから引用する。

```
\def\@aketatekemark#1{%
  \hbox to\z@\{hskip-.25zw\raise.9zh\hbox{%
    \kanjiskip=0pt plus.5pt minus.4pt\added by SF 1998/11/06
    \hbox{\scriptsize \chunberf{#1}}\hsj}}
```

はみだし可能にするには、次に示すように、横幅 0pt とする処理を該当する命令に追加すればよい。次のソースリストをみるとわかるように、単に一行を付け加えただけである。

```
\def\@aketatekemark#1{\leavevmode
  \vbox to0pt{\vss
    \hbox to\z@\{hskip-.25zw\raise.9zh\hbox{%
      \kanjiskip=0pt plus.5pt minus.4pt\added by SF 1998/11/06
      \hbox{\scriptsize \chunberf{#1}}\hsj}}
```

変更前後のソースリストを比較すると、TEX の利便性があきらかになる。TEX はプログラミング言語としての性格をもつので、既存のマクロをもとに改変して新しい機能を追加することができる。今回は自作のマクロの改変をおこなったが、世の中には多くの参考になるマクロパッケージが出回っている。このようなマクロのソースリストは原則としてフリーなので、最小限の約束ごと（先行者の達成を尊重してきちんと引用することなど）をまれば、再利用することが可能である。

## 八・二 条件文

TEX のプログラミング言語としての性格は、条件文が使えることによりあらわれている。本法の長短を判断する \ifdim や整数の大小を判断する \ifnum などが典型的なものである。

条件文を使う例として、西暦年号を和暦年号に変換する \SeirekiToWareki 命令を作ってみよう。ソースリストは次の通りである。

```
\newcount\TempCnt
\def\SeirekiToWareki#1{\TempCnt=#1
\ifnum\TempCnt>1988\advance\TempCnt by -1988\relax
#1\the\TempCnt\relax
\else\ifnum\TempCnt>1925\advance\TempCnt by -1925\relax
#1\kanji\the\TempCnt\relax
\else\ifnum\TempCnt>1911\advance\TempCnt by -1911\relax
#1\kanji\the\TempCnt\relax
\else\ifnum\TempCnt>1867\advance\TempCnt by -1867\relax
#1\the\TempCnt\relax
\else\kanji\the\TempCnt
\fi\fi\fi#1#}
```

#1 の \TempCnt と同じカウンタレジスターを用意する。これに引数 #1 を代入すると、四則計算ができるようになる。 \advance 命令計算する。 \ifnum による条件文により、一九八八年より大きい数は平成となる。これを引いて年号にする。昭和、大正、明治への換算も同様にしておこなう。 \the 命令はカウンタの数値を

47 TEX/LAT<sub>E</sub>X は、多くのプログラミング言語から初めて発展してきたものである。すなわち機能をもちパッケージを利用する以前は、作者のプログラミング能力に強く依存していたのである。



次の例は、「字数 × 行数」という設定をおこなった例である。

```
\textwidth=37zw
\baselineskip=24\N
\topskip=0.5zw
\textheight=39\baselineskip
\advance\textheight by1zw %版面を守るために微調整
\settopoint\textheight
\maxdepth=opt
```

余白優先の設定もできる。実際、articleクラスなどの $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 標準の文書形式では、クラスファイルの中で計算を行ない、余白優先の設定がおこなわれている。上述の版面優先の設定で充分であると考えられるので、ここでは余白優先の設定の方法の説明は省略する。

#### 版面の原点

左上方が基準点になっており、(D) $\text{L}^{\text{A}}\text{TEX}$ の標準では、右方向にインチの余白、右方向にインチの余白が取られるようになっていいる。つまり、左上方から (lin, lin) のところが版面の原点になる。この原点からの差分で余白を設定することができる。次に、設定例をあげる。

```
\topmargin=-0.44cm%
\oddsidemargin=11.25mm
\evensidemargin=11.25mm
```

#### 文字の配置

縦組の文字は、仮想ボディーの中心が基準線となる。これと組み合せる欧文書体の基準線は自動的に設定される。この基準線を動かすには、`\tbaselineshift` というパラメーター (寸法レジスター) の値を変更する<sup>51</sup>。

`\raisebox` 命令を使えば、文字を送りの方向に  動かすことができる。たとえば、`\smash to \raisebox{1zw}{\fbox{す}}` と入力したものは、`\smash to`、文字列の横幅を奪う命令である。これによって、固五行送りが確保できる。

#### 字送りと字間

$\text{L}^{\text{A}}\text{TEX}$  は、箱とグルーのモデルをとっているので、本来の行整形・段落整形の局面では、必然的に字間の制御をおこなうことになる。手動で字間の制御をおこなうには `\kern` 命令を用いる。たとえば、`すにに` と入力すると、`す` と `に` の間に `\kern-0.75zw` と入力すると、`すに` のように重なった位置に出力される。

$\text{L}^{\text{A}}\text{TEX}$  を使いつづけて、字送りの制御をおこなう局面はない。強いておこなうとすると、たとえば、次の方法が考えられる。

```
\hbox to0pt{\ \hss}\kern2zw
```

あまり意味のある試みとはおもえないが、出力結果は `す` と `に`。最初の `を縦幅 0pt` で出力しているので、参照点はそのまま記号の先頭にある。そこから二文字分だけ送ったところに次の記号の参照点がきている。

$\text{L}^{\text{A}}\text{TEX}$  では、カーニング (`\kern`) のほかにスキップ (`\mskip` と `\vskip`) という概念がある。カーンは文字間を指定した寸法だけ動かすが、スキップは、寸法

51 縦組のときは、欧文書体のスタイルの原点を通る線が、和文の中心線「|」が、`す` の中心調整された。注釋、第 147 頁 2 $\epsilon$  附録・縦組編」の第二十三節を参照してください。  
52 本節では、 $\text{L}^{\text{A}}\text{TEX}/\text{L}^{\text{A}}\text{TEX}/\text{X}\text{L}\text{A}\text{T}_{\text{E}}\text{X}$  などのソフトを動かす際に、`makeindex` の種のソフト機能とカーニングを利用してください。

53 本報告の和字は、十三級である。行送りを二番にした場合、はみだし許容としておくべからずが重なる。はみだし許容でない場合は、`\jot`を減らした`TeX`の機能がある。行送りが狂うことのため、このため、はみだし許容の場合は、行送りを充分大きくしておく必要がある。

指定だけでなく延び量・縮み量を指定する`\jot`がある。ただし、`\hskip1pt plus0.1pt minus0.2pt`などの`\dimen`設定も`\jot`がある。これは、`TeX`のルーの考えと関係している。和字と和字の間に挿入される`\kanjiskip`に、延び量・縮み量を設定して行の調整に使う方法についてはすでに詳しく述べた。

#### 行送りと行間

頭注<sup>43</sup>を参照。`TeX`の行送りは、`\baselineskip`というパラメーター（スキップレジスター）で設定する。本報告の場合は二十四函（`24\q`）である。行間の最小値を`\lineskiplimit`（本報告では`Opt`）に設定している。`TeX`は、段落整形のとき`\jot`、前の行の深さを`\prevdepth`に保存している。したがって、`\prevdepth`の値、現在の行の高さ、`\jot`、`\lineskiplimit`の値を加えたものが、行送り`\baselineskip`の値を超える。 `TeX`の処理では、最後の行が重なったとみなされる。このときは、行間`\lineskip`に設定された空きを追加して、前後の行が重ならぬ`\jot`になる。

`\baselineskip`の値は、書体の大きさによって別々に指定する。切り替えは`\Large`や`\small`などの命令の中でおこなうことになるので、書体の大きさを変更すると同時に、行送りもかわる。

#### マスターページのパターン

`TeX/LaTeX`にはマスターページというものはない。かわりに、文書クラスファイルがあり、そこでその文書特有の種々のパラメーターを設定する。これらの文書クラスファイルを文書ファイルに読み込んで、`TeX/LaTeX`の処理にかける。ある文書クラスファイルの設定の一部分を変更したいときは、その設定を文書ファイル（あるいは別ファイル）に書き込んで上書きする。

本報告の組版は、`sftbdk.dcs`という文書クラスファイルをあらたに作成してもちいている。この文書クラスファイルでは、自動ノンブルのみならず自動柱出力（`\section`命令で設定した見出しが自動的に柱に出力される機能）もおこなっている。さらに、試し組版時の柵組格子（グリッド）の出力もおこなえるようになっている。

#### 九・三 改ページと改段

##### ページ末と段末の禁則

これらはすでに述べたとおりである。それぞれペナルティーを設定して、組版に反映させる仕組みになっている。行送り固定の場合は、クラブ行やウィドウ行の禁制はゆるめたほうがよいこともすでに述べた。

#### 九・四 行末処理と改行

##### 行末処理五通りに対応できるか

PREMedia, Part 15, 11001年六月、六五ページに示されたものうち、単一行末揃えの原則（一つの約物の行末形態は一つ）から、二つの方式を選択した（第四・一項参照）。あと三つの方法は、`PDF`では実現がむずかしいが、強いて対応する必要はないと考えている。というのは、同じ約物の処理について、二または三通りのアルゴリズム（ぶら下げ、半角アキ、行末揃え）を考えなければならず、処理自体、簡明でないからである。

54 行末にそのほかに和欧文混植による半端な空きを生ずるので、実際には、`seal`多くの場合分けをしなければならぬ。





60 第三・三項まで、見かけ上、一行の  
 行見出しになっていることがあ  
 る。これは、description 環境  
 の見出しである。二行目以降が、  
 文字が揃っていない。\\subsection  
 による形態で異なることに注意して  
 いる。頭注の参照。なお、次の第十一  
 項目の description 環境の  
 である。

## 九・六 行どり対応

本文中の行どりについては、第七節を参照。別行見出しの行どりに  
 行どり見出し については第七・四項を参照。本報告の \\section の見出しは、二行

中央どりの一行別行見出しを出力するようになってくる。本報告の \\subsection  
 の出力は、二行同行見出しになっている。二行中央どりの二行同行見出し（小書）を  
 出力する工夫が、拙著「続 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 階梯・縦組論」の第十二章に記載されている。

## 十 便利な機能

### 十・一 補助ファイルの利用

DTP ソフトウェアなどを用いる場合、アップルスクリプトなどを用いて目次や索  
 引の作成などおこなうことが多い。この種の機能は、(D) L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> にデフォルトで  
 備わっている。代表的な機能を次にあげる。

見出しの自動出力 (D) L<sup>A</sup>T<sub>E</sub>X では、見出しを自動出力する機能がある。たった一  
 行 \\tableofcontents 命令を付け加えれば、\\section や \\subsection に設  
 定した見出し項目を抽出して、目次の形に整形したものを出力する。本報告の  
 末尾はこのようにして出力したものである。拙著「L<sup>A</sup>T<sub>E</sub>X 本づくりの八衢」第  
 二十章で横組の目次について説明したが、そのまま縦組の目次についても応用  
 できる。本報告の目次は、三点リーダーを用い、見出しの字どりを自動的にお  
 こなうことができるように変更をくわえてある。目次の項目は、toc ファイル  
 に書き出されているので、これを sed や awk などのスクリプト言語で加工する  
 ことももちろん可能である。

索引の作成 索引の項目の抽出は \\index 命令を使うことによる。索引の項目  
 は、idx ファイルに書き出されているので、これをスクリプト言語で加工するこ  
 とが可能である。和文索引、英文索引など複数の索引を作ることでもできる。同  
 掲書第二十一章を参照されたい。

文献リストの作成 文献のデータベースから必要な文献を取り出してリストをつ  
 くり、さらに相互参照をおこなう機能がある (B<sup>B</sup>L<sub>A</sub>T<sub>E</sub>X)。論文や学術書の作成  
 に便利である。詳しくは、拙著「化学者・生化学者のための L<sup>A</sup>T<sub>E</sub>X—パソコン  
 による論文作成の手引」(東京化学同人、一九九三)第十一・十九章を参照。

### 十・二 マークアップ言語との連携

L<sup>A</sup>T<sub>E</sub>X と HTML 筆者は、第二十三回情報化学討論会(二〇〇〇)の世話人を  
 した。このとき、マンストラクトの予稿集印刷とインター  
 ネット配信をおこなったが、L<sup>A</sup>T<sub>E</sub>X と HTML (Hypertext Markup Language) を  
 連携させる手法を応用した。これは、L<sup>A</sup>T<sub>E</sub>X 処理のユーザ、HTML ファイルを同時  
 に出力させるという方法である。

### L<sup>A</sup>T<sub>E</sub>X と HTML/XML

L<sup>A</sup>T<sub>E</sub>X と SGML (Standard Generalized Markup  
 Language) との連携について、詳しくは、拙著

「化学者・生化学者のための L<sup>A</sup>T<sub>E</sub>X」第二十一章を参照。SGML ファイルを L<sup>A</sup>T<sub>E</sub>X  
 ファイルに変換するプログラムは、すでにフリーで公開されているので、これを利用



五・二 高さ(横幅)Opt の箱	15
六 特殊な組版	15
六・一 漢文の訓点文	15
六・二 数式	15
六・三 化学構造式	16
七 行どり	16
七・一 縦組における行どりの重要性	16
七・二 行どり命令	17
七・三 別行立て数式の行どり	17
七・四 別行見出しの行どり	18
七・五 特殊な見出し	18
八 組版プログラミング言語としてのTEX	19
八・一 カウンターと注番号	19
八・二 条件文	20
九 検証	21
九・一 寸法の単位	21
九・二 版面と余白との関係	21
九・三 改ページと改段	23
九・四 行末処理と改行	23
九・五 行中の調整	24
九・六 行どり対応	26
十 便利な機能	26
十・一 補助ファイルの利用	26
十・二 マークアップ言語との連携	26
十一 おわりに	27

## 付 録 (一)

本報告は、原稿作成時には、枘組格子(グリッド)を出力して行送りをチェックし、最終段階で枘組格子を消して出力している。この切り換えは `\pagegridon` 命令および `\pagegridoff` 命令により行う。参考のため、次ページ以降に `\pagegridon` を宣言して、枘組格子を出力したものを掲げる。この部分の設定は次の通りである。

```
\kanjiskip=0pt plus .125zw minus 0.03125zw
\xkanjiskip=.25zw plus .25zw minus .0625zw
```

例文は、青空文庫よりえた「坊ちやた」(夏目漱石)である。句読点は、ふら下げなしで組版している。ルビの部分は、`furikana` パッケージおよび `furikant` パッケージの振りかな命令をもちいて出力したものである。なお、青空文庫では、「無鉄砲《むつぽつ》」のむつにルビを地の文のなかに割注で埋めこみ、拗音・促音を小書にしている。この例文では、ルビの拗音・促音は小書したものをそのまま用いた。特殊な調整をおこなっているところには説明のための頭注を付けた。

63 青空文庫書誌事項―(著者名)夏目漱石、(書籍名)坊ちやた、(底本)ちく

本日本文全集 夏目漱石、筑摩書房、一九九二(平成四)年二月二〇日第一刷発行、(底本の親本)「夏目漱石全集」ちくま文庫、筑摩書房、(入力者名)真先芳秋、(校正者名)柳沢成雄

64 この例文のルビは、おおよそ次の田安に

振って、(一)親文字の間はひらがなで割る。(二)中シキルビを優先する。(三)すくなくともルビの各文字の二分割、該語の親文字にかかるとして許容する。(四)ルビには隣の文字でかかるとして許容する。ただし、ルビのなご漢字には隣の文字のルビはかけない。(五)均等ルビ(グループルビ)では、ルビの文字が隣の親文字にかかっても許容する。(六)行末、行頭に字余りのルビがきた場合は、版面をまもるために細いなご(二割)を最優先にした二割の読みあわせルビに間にあわせしめて適切な仕上がりになっていると見えなごに合わせる。













付 録 (二)

スペースファクターについての補足

第二・一項に述べたスペースファクターが、実際の組版における空きをどのように制御しているかは、`TeXbook`の第十一章に詳しく説明されている。補足の説明を以下に加える。

`Oh, oh!` cried Baby Sally.

とこの例文(文字の大きさを10pt)の空きについて、次のような値があたえられている。

空きの位置	基本量 pt	伸び量 pt	縮み量 pt	備考
<code>Oh,</code>	3.33333	2.08331	0.88889	半角コマのつなぎ
<code>oh!</code>	4.44444	4.99997	0.37036	一重引用符のつなぎ
<code>cried,</code>	3.33333	1.66666	1.11111	単語間
<code>Sally.</code>	4.44444	4.99997	0.37036	半角コマのつなぎ

コマのつなぎに関しては、単語間の空きを基準に計算すれば、次のように計算ができる。伸びと縮みの比のとり方が逆であることに注意。えられた数値1.250を千倍したものが、`\frenchspacing`とコマにあたる値になることがわかる。

$$\frac{2.08331}{1.66666} = 1.250 \quad \frac{1.11111}{0.88889} = 1.250 \quad (1)$$

コマのつなぎに関しては、単語間の空きを基準として、同様に計算するということができる。えられた数値3.000を千倍したものが、`\frenchspacing`とコマにあたる値であることがわかる。

$$\frac{4.99997}{1.66666} = 3.000 \quad \frac{1.11111}{0.37036} = 3.000 \quad (2)$$

悪評価についての補足

第二・二項で述べた悪評価値 (badness) の計算例を示す。計算した値が10000を超える場合は、10000とする。

- 1 (詰め) 自然に組版したときに縮み量の合計が10ptのものを9ptまで縮めて一行に収める場合には、次のように計算する。

$$b = 100 \times \left(\frac{9}{10}\right)^3 = 73 \quad (3)$$

- 2 (延ばし) 自然に組版したときに伸び量の合計が10ptのものを20ptまで延ばして一行に収める場合には、次のように計算する。

$$b = 100 \times \left(\frac{20}{10}\right)^3 = 800 \quad (4)$$

幅なしの箱の実現方法についての補足

第四・三項で述べたものを含めて、幅なしの箱の実現方法には、次に示すようにいろいろなものがある。行末にきた`\hskip`などによるグループは、単独で捨てられ

