pTEX 2.1.5 における数式の前後

中野 賢*

1997 年 8 月 8 日

はじめに

pTEX では、和文と欧文の間に**四分アキ**と呼ばれるスペースを自動的に挿入するようになっている。挿入される箇所は、和文と欧文の間というだけでなく、他にも次の条件が揃っていなくてはならない。

- `\autoxspcode` の状態
- `\xspcode` および `\inhibitxspcode` による抑制を受けない
- shift_amount(移動量) がゼロである `\hbox` の直前、直後
- 合字、ペナルティ、ベースライン調整量、暗黙の `\kern` の直前、直後

本文では多くの場合、単純に文字が並んでいるだけなので、このような条件を知らなくても、期待している結果を得ることができる。しかし、数式を組み立てた場合、自分ではとくに指定をしていなくても、TEX が内部処理で、文字の位置を調整したり、`\vbox` に入れたりするために、四分アキが入らず、その原因もわからないということがある。

そこで、pTEX 2.1.5 では、テキスト数式の前後への四分アキの挿入方法を変更した。この文書では、その拡張に際しての仕様の変更について簡単に説明をしている。

1 pTEX 2.1.4 までの数式の前後

pTEX 2.1.4 で[1]、テキスト数式の前や後に四分アキが入らない主な原因は以下の 3 つである。

- `\xspcode` による影響の場合

*Ken Nakano (<ken-na at ascii.co.jp>)：株式会社 アスキー 出版技術部

[1] より正確に言えば、pTEX 2.1.5 β 6 まで

- シフトされた \hbox となる場合
- \vbox となる場合

1.1 \xspcode による影響の場合

\xspcode および \inhibitxspcode による抑制は、たとえば、α や β の前後にスペースが入らないという結果をもたらす。なぜならば、α は cmmi10 の "0B の位置、β は "0C の位置にあるからである。これら文字の \xspcode の値はゼロ、すなわち前後への四分アキを抑制するという動作をする。その結果、前後に四分アキが入らないのである。

1.2 シフトされた \hbox となる場合

shift_amount がゼロでない \hbox の例としては、$\sum$ や $\int$ や x^2 や x_2 や $\sqrt{\mathrm{Var(X)}}$ などを挙げられる。

$\sum$ は cmex10 の "50 の位置にあり、これは cmr10 の “P” と同じ位置なので \xspcode による抑制は受けない。しかし、位置の調整のために、 \hbox に入れられ、左にシフトされる。この結果、 "50 の文字が一つだけにも関わらず、処理対象外の \hbox となってしまい、前後への四分アキが入らなくなる。$\int$ も同様である。

x^2 や x_2 などの場合は、上付き/下付き文字がシフトされた \hbox として組み立てられているので、“x” の前には四分アキが入るが、“2” の後ろには入らないことになる。

根号記号の場合は、全体がシフトされた \hbox として組み立てられるので、前後に四分アキが入らない。

1.3 \vbox となる場合

その他に四分アキが入る箇所は、合字、ペナルティ、ベースライン調整量、暗黙の \kern の直前か直後だけである。これ以外の箇所には入らない。数式の前後に限れば、このうちの、 \vbox の前後には入らないという制限が大きく影響をしている。たとえば、$\frac{1}{k}$ といった分数や、x_k^2 のように上付きと下付き文字の両方を指定したときが、これに当たる。

分数の場合は、最終的に \vbox として組み立てられる。そのため、分数の前後に四分アキが入らなくなる。上付きと下付きの両方がある場合は、それらが \vbox に入れられるため、入らなくなるのである。上付きと下付きの一方の場合と状況は異なるが、結果は一緒である。

2　pTEX 2.1.5 $\beta 7$ での数式の前後

今までの実装コードでは、数式の開始や終了のノードを見つけても、開始/終了ノードを単純にスキップするだけで、その後、数式の内部の文字を見て、スペースを挿入するかどうかを判断をしていた。そのために数式を囲んでいる \hbox がシフトしていたり、\vbox であったりすると前後にスペースが入らないという結果になっていた。

そこで、$\beta 7$ では、数式の開始ノードを見つけたとき、次の要素が何であるかを調べることにした。

数式の次の要素が “文字” そのものである場合、テキストの最後の文字との関係で適切なスペースを入れる。ここで、従来の動作と異なるのは、数式内部の文字の \xspcode あるいは \inhibitxspcode の設定を無視するようにした点である。つまり、数式内の文字とその直前との文字が和文か欧文かだけを調べて、四分アキか漢字間スペースを挿入するようにした。この変更で α や β などの文字の前にスペースが入るようになる。

数式の開始ノードの次の要素が文字以外の場合は、それが何であっても四分アキを入れる。この修正によって、数式がシフトされた \hbox や \vbox の場合であっても四分アキが入る。このようなボックスになるときは、上付き/下付き文字、分数、数学記号を使った場合などである。

終了ノードの時点では、この後に挿入するスペースが決まっていればそれを用いる。すなわち、数式の最後とその後るの文字がともに和文であれば漢字間スペースを挿入し、いずれかが欧文であれば四分アキを挿入する。このときも、数式内部の最後の文字の \xspcode も無視するようにし、α や β などの後ろにも入るようにしている。

シフトされた \hbox や \vbox などで終了したときは、前側に四分アキを入れたので、合わせる意味で後側にも四分アキを挿入する。これで x^2 や $\frac{dx}{dy}$ や $\sqrt{3}$ などの後ろにもスペースが入る。

なお、数式の直前の文字と数式内部の先頭文字がともに欧文の場合、あるいは数式内部の最後と数式直後がともに欧文の場合は、TEX と同じであり特別なスペースを入れないのは従来と同じである。

3　pTEX 2.1.5 $\beta 8$ での数式の前後

pTEX 2.1.5 $\beta 8$ では、単純にテキスト数式の前後に四分アキを入れる方法で実装している。ただし、「(」や「)」などによる抑制を効かせるべき箇所には四分アキは入らない。つまり “（$y=a+b$）” は “（$y = a + b$）” となり “（ $y = a+b$ ）” とはならないことに注意する。‘（$y=a+b$）’ は ‘（$y = a+b$）’ となり ‘（ $y = a + b$ ）’ とはならないことに注意する。

$\beta 7$ 版と $\beta 8$ 版との仕様の違いは、次のように数式内の文字が漢字で始まる

（あるいは終わる）ときに現れる。

```
\kanjiskip=4pt \xkanjiskip=8pt
□□$表面積=4\pi r^2$□□
```

この例をそれぞれで処理した場合、

□ □ 表 面 積 $= 4\pi r^2$ □ □ (β7 版)
□ □ 表 面 積 $= 4\pi r^2$ □ □ (β8 版)
□ □ 表 面 積 $= 4\pi r^2$ □ □ (この版)

となる。つまり、β7 版では数式内部の文字と直前の文字を比較しているため、前側に漢字間スペースが入り、後側に四分アキが入る。一方、β8 版では数式内部の文字に関係なく、両側に四分アキが入る。

なお、両者の違いは、次のように \hbox を用いたときにも現れる。

```
\kanjiskip=4pt \xkanjiskip=8pt
□□$\hbox{表面積}$□□
```

これを処理すると、β7 版は目指すべき仕様とは異なり、後ろに何も入らない（バグ）。β8 版では仕様どおりに両側に四分アキが入る。

□ □ 表 面 積□ □ (β7 版)
□ □ 表 面 積 □ □ (β7 版を修正した場合)
□ □ 表 面 積 □ □ (β8 版)
□ □ 表 面 積 □ □ (この版)

実際には、pTEX, pLATEX のデフォルトの設定では \kanjiskip と \xkanjiskip の値はもっと小さいので、違いはそれほど目立たないかもしれない。しかし、縦組にした場合は、つぎのような結果となる。

△△表面積 $= 4\pi r^2$ △△ (β7 版)
△△表面積 $= 4\pi r^2$ △△ (β7 版を修正した場合)
△△ 表面積 $= 4\pi r^2$ △△ (β8 版)
△△ 表面積 $= 4\pi r^2$ △△ (この版)

β7 の結果を見ると文字間のスペースは仕様どおりになっている。しかし、ベースラインの調整量が数式の後ろで戻っていないというバグがある。たとえバグを修正したとしても、決して読みやすいとは思えない。

以上の結果を比較すると、β8 の挿入方法のほうが自然で、読みやすいように思える。また、挿入されるスペースも前後の文字、数式内の文字に関係なく、四分アキと決まっているので結果がわかりやすい。そこで、β8 では、この節の冒頭で述べた、

単純に、テキスト数式の前後に四分アキを入れる

という仕様で実装している。

3.1 別の問題

β7 の実装では、完全に数式にだけ影響するようにしていなかったため、通常の文章内でのボックスに対する動作も変わってしまっていた。たとえば、

```
□□\AA △△△△\hbox toOpt{ABC\hss}○○○○\hbox{}A\hbox{}□□□□
```

が

□□ Å △△△△ ABC○○ A □□□□

となる。先ほどのリストを処理して期待する結果は、次のようだと思われる。

□□ Å △△△△ ABC○○A□□□□

この版では、

□□Å △△△△ ABC○○A□□□□

となる。Å の前に四分アキが入っていなければ、それはシフトされたボックスの前後にアキが入らないからである。β7 の実装で “˚” と “A” の位置がずれているのは、シフトされたボックスの前には入れないのは変わらないけれども、後ろで必ず入れないようにするのをやめたためである。また、β7 では、ボックスの中身の最後が文字以外でも、空でも四分アキの挿入に影響をさせないようにしていたため、幅がゼロのボックスを作成しても、参照点の位置が戻らないし、処理の抑制もされなくなっていた[2]。

A 自動挿入されるスペースの種類

pTEX では、日本語文書をきれいに組版するために、和文と和文の間、和文と欧文の間に自動的にスペースを入れるように拡張している。欧文と欧文の間は、TEX のメカニズムそのままである。

和文と和文の間に入れるスペースの量は、`\kanjiskip` という長さレジスタに設定をする。和文と欧文との間は `\xkanjiskip` という長さレジスタである。pTEX のデフォルトでは、

[2] これらの問題は β11 で修正。

```
\kanjiskip=0pt plus .4pt minus .4pt
\xkanjiskip=.25zw plus 1pt minus 1pt
```

となっている。この設定は、\kanjiskip は標準でゼロポイント、場合によって ±0.4 ポイント分だけ伸縮、\xkanjiskip は標準でその時点の和文フォントの幅の 1/4、場合によって ±1 ポイント分だけ伸縮しても良いということを意味している。

\kanjiskip も \xkanjiskip も、段落の終わりか、\hbox の最後の時点の値が有効となる。したがって、ひとつの段落内や \hbox 内で複数回指定をしても、その最後の指定によって処理される。

ただし「、」や「(」のように、特定の文字が連続する場合、そのまま全角幅で並べ、間に \kanjiskip を挿入すると文字の間が離れすぎてしまう。このときには、\kanjiskip ではなく、JFM(Japanese Font Metric) で設定されているスペースの量が使われる。

\xkanjiskip に関しても、和文と「;」、「(」と和文、欧文と「。」、「…」と前後の欧文のような箇所には、スペースを挿入しないほうがきれいに見える。そこで、特定の欧文の文字に対して、\xkanjiskip の挿入を制御するために \xspcode が用意されている。特定の和文に対しては \inhibitxspcode を用いて制御する。

B　スペースに関するプリミティブ

pTEX で拡張した、スペースに関するプリミティブは以下のとおり。

B.1　\kanjiskip, \autospacing, \noautospacing

\kanjiskip は、漢字と漢字の間に自動的に挿入するスペースの量を格納する長さレジスタである。

\autospacing と \noautospacing は、漢字と漢字の間にスペースを挿入するかどうかを指定するのに用いる。\autospacing を指定すると自動的に挿入される。\noautospacing を指定すると漢字間へのスペース挿入は抑制される。

B.2　\xkanjiskip, \autoxspacing, \noautoxspacing

\xkanjiskip は、漢字と英字の間に自動的に挿入するスペースの量を格納する長さレジスタである。

\autoxspacing と \noautoxspacing は、漢字と英字の間にスペースを挿入するかどうかを指定するのに用いる。\autoxspacing を指定すると自動

的に挿入される。 \noautoxspacing を指定すると漢字間へのスペース挿入は抑制される。

B.3 \xspcode

\xspcode は、指定した英字と漢字との間のスペース挿入をどのように**許可**するかの設定をするプリミティブである。動作は、つぎのいずれかの数値で指定をする。

0 前後の漢字との間へのスペースの挿入を禁止する。
1 直前の漢字との間にだけスペースの挿入を許可する。
2 直後の漢字との間にだけスペースの挿入を許可する。
3 前後の漢字との間にスペースの挿入を許可する。

初期値は、[0-9A-Za-z] は３、それ以外はゼロになっている。ただし、以下の文字については、kinsoku.tex で別の値に初期化されている。括弧内は ASCII 文字コード（16 進数）である。

(=1 ("28)　　)=2 ("29)　　[=1 ("5B)　　]=2 ("5D)
‘=1 ("60)　　’=2 ("27)　　;=2 ("3B)　　,=2 ("2C)
.=2 ("2E)

ただし、\xspcode の設定は、文字コードに対してであり、フォントによって異なる値を指定することはできない。したがって "60 の位置にある文字は、cmr10 では “`”, cmmi10 では “ℓ”, cmex10 では “$\coprod$” であるが、これらの文字はすべて \xspcode=1 として処理される。

B.4 \inhibitxspcode

\inhibitxspcode は、指定した漢字と英字の間のスペース挿入をどのように**抑制**するかの設定をするプリミティブである。動作は、つぎのいずれかの数値で指定をする。

0 漢字と英字との間のスペースの挿入を禁止する。
1 直前の英字との間のスペースの挿入を禁止する。
2 直後の英字との間のスペースの挿入を禁止する。
3 前後の英字との間のスペースの挿入を許可する。

初期値は、すべての漢字について３である。ただし、以下の文字については、kinsoku.tex で別の値に初期化されている。

、=1	。=1	，=1	．=1	；=1	？=1
（=2	）=1	［=2	］=1	｛=2	｝=1
‘ =2	’=1	“ =2	”=1	〔=2	〕=1
＜=2	＞=1	《=2	》=1	「=2	」=1
『=2	』=1	【=2	】=1	－=0	～=0
…=0	￥=0	°=0	′=1	″=1	