

Japanese Typesetting with LuaT_EX

Hironori KITAGAWA

12 October 2012

Introduction and outline

We are developing the *LuaTeX-ja* package¹, which offers typesetting Japanese with LuaTeX.

Outline:

- Japanese typesetting rules
- pTeX engine and LuaTeX-ja
- **Vertical** Japanese typesetting

¹<http://sourceforge.jp/projects/luatex-ja/wiki/FrontPage%28en%29>. Most documents are written in Japanese.

“LuaT_EX is a Unicode-aware T_EX engine.
So, does simply using Japanese fonts
mean typesetting Japanese?”

“LuaT_EX is a Unicode-aware T_EX engine.
So, does simply using Japanese fonts
mean typesetting Japanese?”

→ **No!**

Terrible output by LuaTeX

Input

```

1 \fontspec[Ligatures=TeX]{ipaexm.ttf}
2 % proportional (IPAexMincho)
3 今は東海道新幹線における「のぞみ」
4 「ひかり」「こだま」の3種別とも`Super express' '%
5 (超特急) とアナウンスされている。

```

Output

今は東海道新幹線における「のぞみ」	「ひかり」
per	express' ' (超特急)
とアナウンスされている。	

This is terrible ...

Japanese typesetting rules

The next four slides describe basic rules of Japanese typesetting.

- 1 Most basic principles**
- 2 Punctuation marks and glue around alphabets²**
- 3 Prohibited line-breaking**
- 4 Justtification**

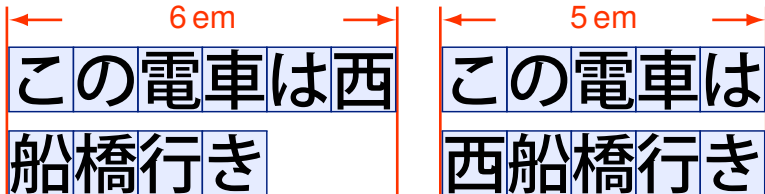
²In this talk, “alphabets” means that alphanumeric characters.

Basic principles

- 1 Most characters have square bounding box.
- 2 These characters are typeset solid.
The line length should be a multiple of em.
- 3 There is no inter-word space.

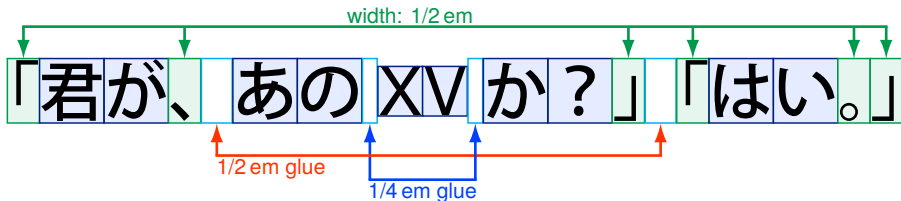
Sample text

この電車は西船橋行き



Punctuaton marks

- 1 Width of some Japanese punctuation marks are $1/2$ em, accompanied with some glue.
- 2 Usually $1/4$ em glue is inserted between Japanese characters and alphabets.



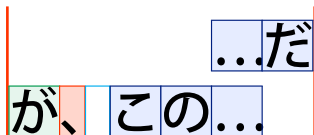
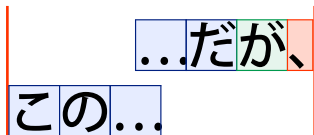
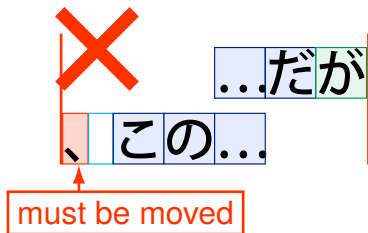
Prohibited line-breaking

1 No line should begin with

- Closing brackets (`)`, `]`, `】`, `》`, ...)
- Commas (`,` and `、`), full stops (`.` and `。`), ...

2 No line should end with opening brackets

(`(`, `[`, `「`, `【`, ...).



Justification

The following methods are used to justify lines:

- 1 stretch or shrink glues
(around punctuation marks and alphabets)
- 2 insert space between two Japanese characters, if the above does not suffice

Example of justification

moved to the 2nd line

この方針では「
正」と「負」で

Example of justification

moved to the 2nd line

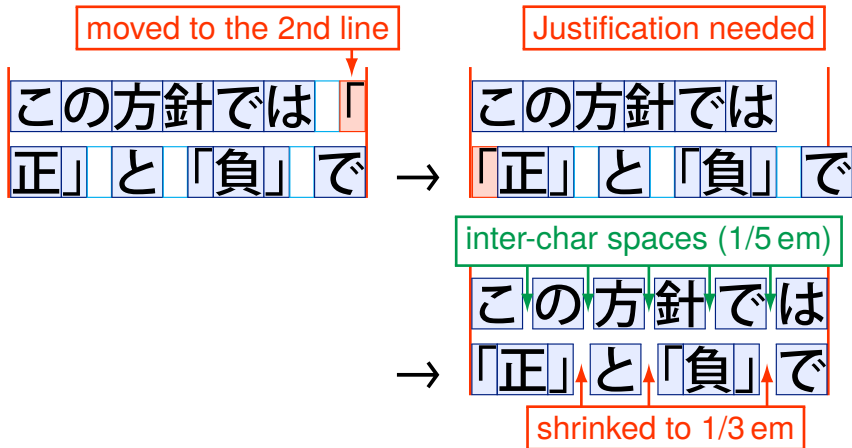
この方針では「
正」と「負」で

→

Justification needed

この方針では
「正」と「負」で

Example of justification



Japanese pTeX

Japanese pTeX (developed by ASCII Corp.) is an **engine extension** of T_EX82.

- tuned for Japanese typesetting rules.
- supports both horizontal writing and vertical writing.
- has been the de facto standard for Japanese T_EX users.

Output by pT_EX

Input (pT_EX)

Inter-char space

```
1 \kanjiskip=0pt plus 0.05em
2 \xkanjiskip=0.25em plus 0.25em minus 0.125em
3 今は東海道新幹線における「のぞみ」
4 「ひかり」「こだま」の3種別とも`Super express' '%
5 (超特急) とアナウンスされている。
```

Output (pT_EX)

今は東海道新幹線における「のぞみ」「ひかり」「こだま」の3種別とも“Super express”
(超特急) とアナウンスされている。

Above output (almost) follows Japanese typesetting rules.

Weak point of pTeX

- Even ε -TeX extension is unsupported.
(But we have a fork of pTeX since 2008.)
- No plan for Omega or pdfTeX extensions.
→ Another fork?

Weak point of pTeX

- Even ε -TeX extension is unsupported.
(But we have a fork of pTeX since 2008.)
- No plan for Omega or pdfTeX extensions.
→ Another fork?

If an existing engine (e.g., LuaTeX) has a potential, we only have to develop a macro package for Japanese typesetting.

Weak point of pTeX

- Even ε -TeX extension is unsupported.
(But we have a fork of pTeX since 2008.)
- No plan for Omega or pdfTeX extensions.
→ Another fork?

If an existing engine (e.g., LuaTeX) has a potential, we only have to develop a macro package for Japanese typesetting.

→ **LuaTeX-ja**

Development policy of LuaTeX-ja

- 1 The quality of outputs by LuaTeX-ja should be as good as (or, better than) that of pTeX.
- 2 A source file for pTeX can be reused by LuaTeX-ja with minimal changes.
- 3 Inconvenient or unwanted features of pTeX will be 'fixed' in LuaTeX-ja.

Output by LuaT_EX-ja

Input (LuaT_EX-ja)

LuaT_EX-ja equivalent
of `\kanjiskip=...`

```
1 \ltjsetparameter{kanjiskip=0pt plus 0.05em,  
2   xkanjiskip=0.25em plus 0.25em minus 0.125em}  
3 今は東海道新幹線における「のぞみ」  
4 「ひかり」「こだま」の3種別とも`Super express'`%  
5 (超特急) とアナウンスされている。
```

Output (LuaT_EX-ja)

Same output

今は東海道新幹線における「のぞみ」「ひかり」「こだま」の3種別とも“Super express”(超特急) とアナウンスされている。

Justification with priority

(Inter-char spaces) = 0 pt plus 0.05 em **minus 0.05 em**

No justification

overshoots about 1/2 em

自然数の公理化は、「ペアノの公理系」(Peano axioms)が知られて...

Justification without priority

自然数の公理化は、「ペアノの公理系」(Peano axioms)が知られて...

- pTeX can justify lines only in this approach.
- Negative inter-char spaces. → There is another way.

Justification with priority

(Inter-char spaces) = 0 pt plus 0.05 em **minus 0.05 em**

No justification

overshoots about 1/2 em

自然数の公理化は、「ペアノの公理系」(Peano axioms)が知られて...

Justification with priority (optional in LuaTeX-ja)

自然数の公理化は、「ペアノの公理系」(Peano axioms)が知られて...

- Try to keep “no inter-char spaces”,
at the cost of other glues.
- Most Japanese think that this is “more natural”.

Justification with priority

No justification

overshoots by 1/2 em

あいうえおさし・すせそ 「『かきく

Justification with priority

あいうえおさし・すせそ 「『かきく

- We shrink glues around the middle dot (●).
- Hence the glue before the corner bracket are unchanged.

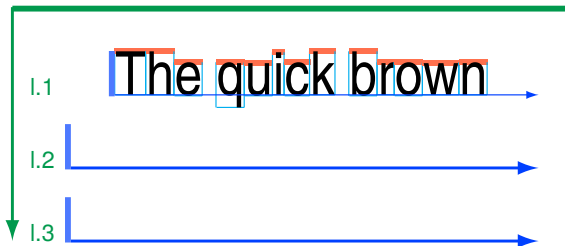
Vertical Typesetting

- **Notations of writing directions**
- **Rules of vertical typesetting**
- **What features are needed in Lua \TeX ?**
 - Usable `xoffset` and `yoffset`
 - RTR direction

TLT direction

In English documents,

- 1 A page begins from **T** (top).
- 2 A line begins from **L** (left).
- 3 The 'top' direction of a glyph is **T** (top).

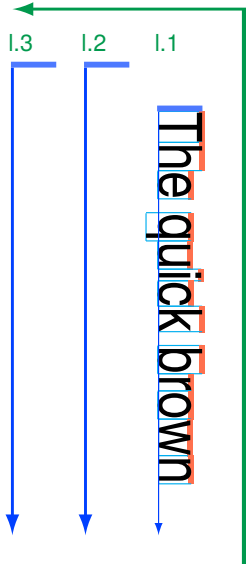


Hence, this direction is denoted by **TLT**.

RTR direction (not in LuaT_EX)

- A page begins from **R** (right).
- A line begins from **T** (top).
- The 'top' direction of a glyph is **R** (right).

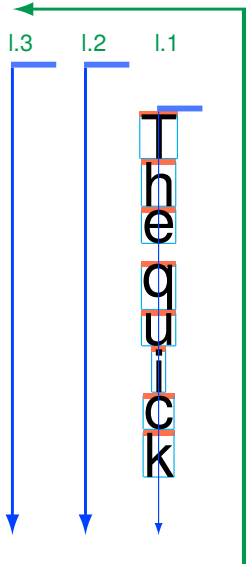
Therefore, alphabets should be rotated 90 degrees clockwise in RTR direction.



RTT Direction

- A page begins from **R** (right).
- A line begins from **T** (top).
- The 'top' direction of a glyph is **T** (top).

In LuaT_EX, characters (and TLT boxes) are **centered around the baseline**.



Example of vertical writing

horizontal writing

(横組み, `\yoko` in `pTeX`)

この十年の間を振り返ると、各種ツールの日本語化が一段落して日本語 `TeX` として `pTeX` がデファクトスタンダードになり、`dvi` から直接 PDF 作成を行う `DVIPDFMx` が広まりました。また、インストールの簡便化を目指した活動が活発となり、新たな日本語符号化……

vertical writing

(縦組み, `\tate` in `pTeX`)

この十年の間を振り返ると、各種ツールの日本語化が一段落して日本語 `TeX` として `pTeX` がデファクトスタンダードになり、`dvi` から直接 PDF 作成を行う `DVIPDFMx` が広まりました。また、インストールの簡便化を目指した活動が活発となり、新たな日本語符号化……

(from 「`TeX` ユーザの集い 2009 予稿集」)

Example of vertical writing

Common usage:

- novels
- newspapers
- comics

A page begins
from **R** (right).

A line begins
from **T** (top).

vertical writing

(縦組み, \tate in p_TEX)

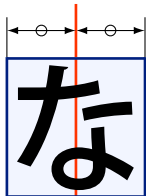
この十年の間を振り返ると、各種ツールの日本語化が一段落して日本語_TEXとしてp_TEXがデファクトスタンダードになり、dviから直接PDF作成を行うDVI_TPDF_Tが広まりました。また、インストールの簡便化を目指した活動が活発となり、新たな日本語符号化……

(from 「_TEX ユーザの集い 2009 予稿集」)

Japanese characters



1



Japanese characters are aligned along their center line.

2

Some symbols have vertical forms.



Ideographic
full stop

Fullwidth left
parenthesis

horizontal
form



vertical
form



Alphabets

center

は

- 1 Alphabets are rotated 90 degrees clockwise.

C

- 2 The baseline of alphabets is shifted.

a



Simple strategy

the horizontal center of 'M'
= that of 'あ'

n

c

a

y

あ

列

Two-digit numbers

It is common for two-digit numbers to typeset horizontally (**Figure 1**).

- Sometimes we supply spaces (**Figure 2**).

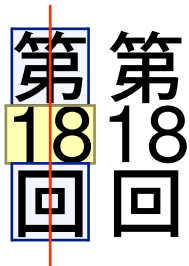


Figure 1

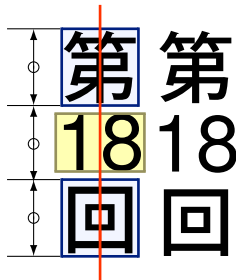


Figure 2

Rules in Lua \TeX 's terms

Now we interpret rules in Lua \TeX 's terms:

Japanese characters RTT direction

alphabets **RTR direction**

shifting baseline

Lua \TeX 's x/yoffset features

two-digit numbers a box in TLT direction



Rules in LuaT_EX's terms

Now we interpret rules in LuaT_EX's terms:

Japanese characters RTT direction

alphabets **RTR direction**

not implemented in LuaT_EX

shifting baseline **need to be fixed**

LuaT_EX's x/yoffset features

two-digit numbers a box in TLT direction



xoffset and yoffset (in TLT)

Lua \TeX can displace a glyph, by modifying `xoffset` or `yoffset` fields of that glyph node.

- 1 `xoffset` doesn't change the bounding box.

$-1/2\text{em}$ $-1/4\text{em}$ 0 $1/4\text{em}$

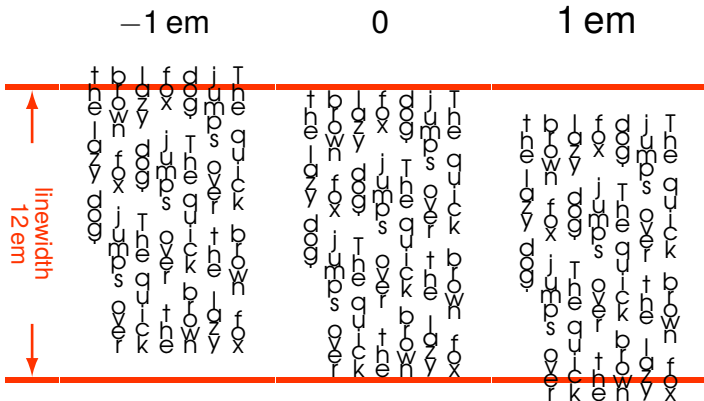
- 2 `yoffset` changes the height and depth.

$-1/2\text{em}$ $-1/4\text{em}$ 0 $1/4\text{em}$

xoffset in RTT

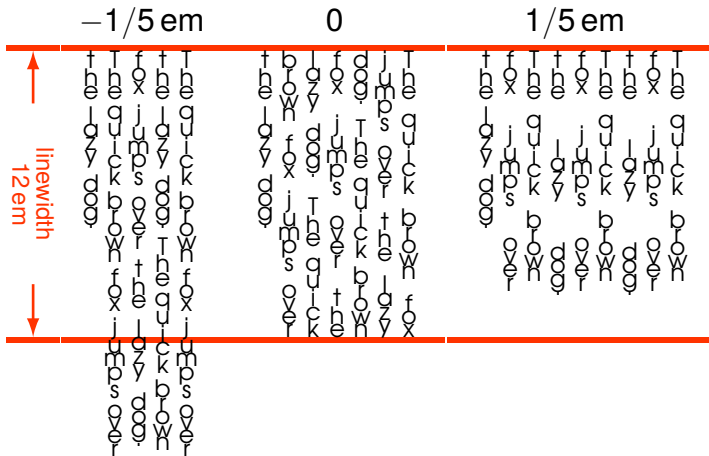
In current Lua \TeX , we cannot use `xoffset` nor `yoffset` to **displace a glyph horizontally in RTT**.

- `xoffset` displaces a glyph vertically.
(This behavior is not so unnatural.)



yoffset in RTT

- 2 yoffset changes the “apparent line width”.

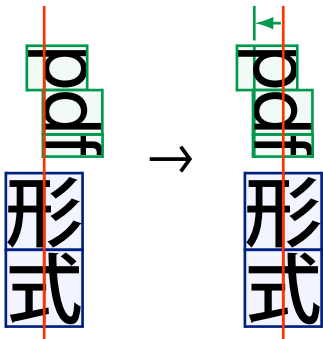


RTR is suitable for alphabets

- **no need for virtual fonts of alphabets**

Rotation of glyphs were handled by “typeset in RTR direction”.

- **simple computation of offset**



The same offset values can be used by any alphabets in a font.

Conclusion

- Using our LuaT_EX-ja package, we can typeset **horizontal** Japanese documents which follows Japanese typesetting rules.
- However, some refinements are needed in LuaT_EX, for **vertical** typesetting:
 - RTR direction
 - Usable `xoffset` and `yoffset`

Conclusion

- Using our LuaT_EX-ja package, we can typeset **horizontal** Japanese documents which follows Japanese typesetting rules.
- However, some refinements are needed in LuaT_EX, for **vertical** typesetting:
 - RTR direction
 - Usable `xoffset` and `yoffset`

Thank you for listening.